

MPMC

Course File

Contents required for course file

1. Cover Page
2. Syllabus copy
3. Vision of the Department
4. Mission of the Department
5. PEOs and POs
6. Course objectives and outcomes
7. Brief notes on the importance of the course and how it fits into the curriculum
8. prerequisites
9. Instructional Learning Outcomes
10. Course mapping with PEOs and POs
11. Class Time Table
12. Individual Time Table
13. Micro Plan with dates and closure report
14. Detailed notes
15. Additional topics
16. University Question papers of previous years
17. Question Bank
18. Assignment topics
19. Unit wise Quiz Questions
20. Tutorial problems
21. Known gaps ,if any
22. Discussion topics
23. References, Journals, websites and E-links
24. Quality Control Sheets
25. Student List
26. Group-Wise students list for discussion topics

1.cover page

<u>GEETHANJALI COLLEGE OF ENGINEERING AND TECHNOLOGY</u>	
<u>DEPARTMENT OF <i>Electronics and Communication Engineering</i></u>	
(Name of the Subject) : Microprocessors and Microcontrollers <u>Course file</u>	
(JNTU CODE – A60430)	Programme : <i>UG</i>
Branch: ECE	Version No : 0
Year: <i>III</i>	Document No: GCET/ECE/56012/01
Semester: II	No. of pages :
Classification status (Unrestricted / Restricted) : Unrestricted	
Distribution List : Dept. Library, Dept Office, Concerned Faculty	
Prepared by	
1) Name : M.Laxmi/ P.Sneha Naga Shilpa/D.Sony/M.V.Sireesha	
2) Sign :	
3) Design : Assoc. Professor/Asst. Professors	
4) Date : 25/11/2015	
Verified by : 1) Name :	* <u>For Q.C Only.</u>
2) Sign :	1) Name :
3) Design :	2) Sign :
4) Date :	3) Design :
	4) Date :
Approved by : (HOD) 1) Name : Dr.P.Sri Hari	
2) Sign :	
3) Date :	

2. Syllabus copy

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY

III YEAR B.TECH.ECE-IISEM

(A60430) MICROPROCESSORS AND MICROCONTROLLERS

Course Objective:

The course objectives are:

- To develop an in-depth understanding of the operation of microprocessor and microcontrollers, machine language programming and interfacing techniques

UNIT-1

8080 Architecture: 8086 Architecture–functional diagram, Register organization, memory segmentation, Programming model, memory addresses, Physical memory organization, Architecture of 8086, Signal description of 8086, Common function signals, Timing diagrams, Interrupts of 8086.

UNIT-II

Instruction set and assembly language programming of 8086: Instruction formats, addressing modes, Instruction set, Assembler directives, macros, Simple programs involving logical, branch and call instructions, sorting, evaluating arithmetic expressions, string manipulations.

UNIT-III

I/O interface: 8255-PPI, various Modes of operation and interfacing to 8086, Modes of operation and interfacing to 8086, Interfacing keyboard, display, D/A and A/D converter.

Interfacing with advanced devices: Memory interfacing to 8086, Interrupt structure of 8086, vector interrupt table, interrupt service routine.

Communication interface: Serial communication standards, serial data transfer schemes, 8251 USART architecture and interfacing.

UNIT-IV

Introduction to microcontrollers: Overview of 8051 microcontrollers, Architecture, I/O ports, memory organization, Addressing modes, Instruction set of 8051, Simple programs.

UNIT-V

8051 real time control: Programming timer interrupts, Programming external hardware interrupts, Programming the serial communication interrupts, Programming 8051 timers/counters.

Course Outcome:

Upon completion of the course:

- The student will learn the internal organization of popular 8086/8051 microprocessors/microcontrollers.
- The student will learn hardware and software interaction and integration.
- The students will learn the design of microprocessors/microcontrollers- based systems.

3.Vision of the Department

To impart quality technical education in Electronics and Communication Engineering emphasizing analysis, design/synthesis and evaluation of hardware/embedded software using various Electronic Design Automation (EDA) tools with accent on creativity, innovation and research thereby producing competent engineers who can meet global challenges with societal commitment.

4. Mission of the Department

- i. To impart quality education in fundamentals of basic sciences, mathematics, electronics and communication engineering through innovative teaching-learning processes.
- ii. To facilitate Graduates define, design, and solve engineering problems in the field of Electronics and Communication Engineering using various Electronic Design Automation (EDA) tools.
- iii. To encourage research culture among faculty and students thereby facilitating them to be creative and innovative through constant interaction with R & D organizations and Industry.
- iv. To inculcate teamwork, imbibe leadership qualities, professional ethics and social responsibilities in students and faculty

5.PEOs and POs

Program Educational Objectives of B. Tech (ECE) Program :

- I. To prepare students with excellent comprehension of basic sciences, mathematics and engineering subjects facilitating them to gain employment or pursue postgraduate studies with an appreciation for lifelong learning.
- II. To train students with problem solving capabilities such as analysis and design with adequate practical skills wherein they demonstrate creativity and innovation that would enable them to develop state of the art equipment and technologies of multidisciplinary nature for societal development.
- III. To inculcate positive attitude, professional ethics, effective communication and interpersonal skills which would facilitate them to succeed in the chosen profession exhibiting creativity and innovation through research and development both as team member and as well as leader.

Program Outcomes of B.Tech ECE Program

1. An ability to apply knowledge of Mathematics, Science, and Engineering to solve complex engineering problems of Electronics and Communication Engineering systems.
2. An ability to model, simulate and design Electronics and Communication Engineering systems, conduct experiments, as well as analyze and interpret data and prepare a report with conclusions.
3. An ability to design an Electronics and Communication Engineering system, component, or process to meet desired needs within the realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability and sustainability.
4. An ability to function on multidisciplinary teams involving interpersonal skills.
5. An ability to identify, formulate and solve engineering problems of multidisciplinary nature.
6. An understanding of professional and ethical responsibilities involved in the practice of Electronics and Communication Engineering profession.
7. An ability to communicate effectively with a range of audience on complex engineering problems of multidisciplinary nature both in oral and written form.
8. The broad education necessary to understand the impact of engineering solutions in a global, economic, environmental and societal context.
9. A recognition of the need for, and an ability to engage in life-long learning and acquire the capability for the same.
10. A knowledge of contemporary issues involved in the practice of Electronics and Communication Engineering profession
11. An ability to use the techniques, skills and modern engineering tools necessary for engineering practice.
12. An ability to use modern Electronic Design Automation (EDA) tools, software and electronic equipment to analyze, synthesize and evaluate Electronics and Communication Engineering systems for multidisciplinary tasks.
13. Apply engineering and project management principles to one's own work and also to manage projects of multidisciplinary nature.

6..Course objectives and outcomes

Course objectives :

1. To understand the basic 8, 16 bit microprocessor architecture and its functionalities.
2. To understand the programming model of microprocessor.
3. To develop the microprocessor based programs for various applications.
4. To make the interfacing in between microprocessor and various peripherals.
5. To develop DOS/BIOS programs.
6. To develop the microcontroller based programs for various applications.
7. To enable the students to understand basic feature of 8051 and AVR controller

course outcomes:

CO 1: Basic understanding of 8085 and 8086 microprocessors architectures and its functionalities.
CO 2: Design and develop 8086 Microprocessor based systems for real time applications using low level language like ALP
CO 3: Interface external peripherals and I/O devices and program the 8086 microprocessor.
CO 4: To write Interrupt service Routine(ISR) to handle interrupts in 8086 microprocessor.
CO 5: To transmit data serially in Multi-processor applications
CO 6: Basic understanding of 8051 microcontrollers architectures and its functionalities.
CO 7: Design and develop microcontroller based systems for real time applications using low level language like ALP
CO 8: To distinguish between RISC and CISC microcontrollers and Analyze the architecture of AVR family microcontroller

7. Brief notes on the importance of the course and how it fits into the curriculum

1. The student will be able to understand the basic microprocessors architecture its functionality.and to interface with I/O devices.
2. This courset will gives the knowledge to understand the advanced processors and its interfacing with various devices.
3. It will be the base for embedded system design.
4. The course will be usefull for understanding computer organization and advanced systems.

8.Pre Requisites: Switching Theory & Logic Design ,Computer Organization.

9.Instructional Learning Outcomes

UNIT – I

1. Student understands the architecture of 8085 and 8086 processor.
2. Student understands the read (input) and write (output) of data to or from the memory.
3. Student gets complete knowledge of registers and memory system of 8086 processor.
4. Student understands the single and multi-processor mode of 8086 processor.

UNIT – II

1. Student understands the various instruction formats of 8086 processor and can convert assembly language code into machine language code.
2. Student understands the various addressing modes required for assembly language programming and can calculate the physical address.
3. Student can develop various assembly language programs in real time applications using microprocessor.

UNIT –III

1. Student understands the architecture of 8255 PPI.
2. Student understands the various command words and modes of operation of 8255 PPI.
3. Student can develop real-time interfacing devices and interface them with 8086 processor for wider applications.
4. Student can develop the interfacing of memory (RAM, ROM, PROM, EEPROM) with 8086 processor.
5. Student understands interrupt structure, interrupt vector table and interrupt service routines etc.
6. Student understands the architecture of 8251USART.

7. Student understands the various command words, modes of operation and interfacing of 8251 USART with 8086 processor.

UNIT –IV

1. Student understands the architecture of 8051 microcontroller.
2. Student understands the addressing modes and instruction set of 8051 microcontroller.
3. Student can develop various assembly language programs in real time applications using microcontrollers.

UNIT –V

1. Student can develop the programming of timers/counters of 8051 microcontroller in real time systems.
2. Student can develop the programming of interrupts of 8051 microcontroller in real time systems.
3. Student can develop the programming of serial communication interrupts of 8051 microcontroller in real time systems.

10.Course mapping with PEOs and POs

Course mapping with PO's

Course Outcomes	Program outcomes												
	1	2	3	4	5	6	7	8	9	10	11	12	13
CO 1: Basic understanding of 8085 and 8086 microprocessors architectures and its functionalities		2		2	2							1	
CO 2: Design and develop Microprocessor based systems for real time applications using low level language like ALP		2		2	2							1	
CO 3: Interface external peripherals and I/O devices and program the 8086 microprocessor.		2		2	2							1	

CO 4: To write Interrupt service Routine(ISR) to handle interrupts in 8086 microprocessor.		2		2	2							1	
CO 5: To transmit data serially in Multi-processor applications		2		2	2							1	
CO 6: Basic understanding of 8051 microcontrollers architectures and its functionalities		2		2	2							1	
CO 7: Design and develop microcontroller based systems for real time applications using low level language like ALP		2		2	2							1	
CO 8: To distinguish between RISC and CISC microcontrollers and Analyze the architecture of AVR family microcontroller		2		2	2							1	

Course mapping with PEOs

	PROGRAMME EDUCATIONAL OBJECTIVES					
	Domain knowledge	Professional Employment	Higher Degrees	Engineering citizenship	Lifelong Learning	Research and Development
a) an ability to apply the knowledge of Mathematics, science and engineering in Electronics and communications	√	√	√	√	√	√
b) an ability to Design & Conduct Experiments, as well as analyze & Interpret Data	√	√	√	√	√	√
c) an ability to design a system, component, or process to meet desired needs with in realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and	√	√	√		√	√

sustainability						
d) an ability to function on multidisciplinary teams	√	√			√	√
e) an ability to Identify, Formulate & Solve problems in the area of Electronics and Communications Engineering	√	√	√		√	√
h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context	√	√	√		√	√
i) a recognition of the need for, and an ability to engage in life-long learning	√	√	√	√	√	√
k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice	√	√	√	√	√	√

12. Class Time Table

Hard copies are attached

13.Micro Plan:

Name of the Faculty **M.Laxmi/P.Sneha Naga Shilpa/D.Sony/M.V.Sireesha**

Branch: **ECE**

Year & Sem: **III-II**

Subject: **Microprocessors and microcontrollers**

Lesson Plan

S.L no	Unit No	Total no of Periods	Topics to be covered	Reg/ Additional	Teaching aids LCD/OHP/BB	Remarks
1	1	11	Overview of basics from Switching Theory, Logic Design and Computer Organization	Additional	BB	
2			Overview of MPMC syllabus and its importance in Real world	Additional	BB	
3			Evolution of microprocessors and Introduction to 8085 microprocessor,	Additional	OHP, BB	
4			Architecture of 8086 Microprocessor – functional diagram description	Regular	OHP, BB	
5			Register organization, memory segmentation	Regular	OHP, BB	
6			Programming model, memory addresses	Regular	OHP, BB	
7			Physical memory organization	Regular	BB	
8			Signal description of 8086, Common function signals	Regular	BB	
9			maximum and minimum mode signals	Regular	BB	
10			Timing diagrams, Interrupts of 8086	Regular	OHP, BB	
11			Tutorial class	Regular	BB	
12	II	08	Instruction formats, addressing modes	Regular	OHP, BB	
13			Instruction set of 8086	Regular	BB	
14			Instruction set of 8086	Regular	BB	
15			Assembler directives, macros	Regular	OHP, BB	
16			Simple programs on logical, branch and call instructions	Regular	BB	
17			Programs on sorting, string manipulations	Regular	BB	
18			Tutorial class	Regular	BB	
19			Introduction to Pentium and dual core processors	Additional	OHP, BB	
20	III	20	Introduction to parallel data transfer methods and importance of port structure	Regular	OHP, BB	
21			8255-PPI	Regular	OHP, BB	
22			Modes of operation and interfacing to 8086	Regular	BB	
23			Modes of operation and interfacing to 8086	Regular	BB	
24			Interfacing keyboard, display to 8086	Regular	OHP, BB	

25			Stepper motor interfacing to 8086	Missing	BB	
26			A/D converter interfacing to 8086	Regular	OHP,BB	
27			D/A converter interfacing to 8086	Regular	BB	
28			Tutorial class	Regular	LCD,OHP,BB	
29			Memory interfacing to 8086	Regular	OHP,BB	
30			Memory interfacing to 8086	Regular	BB	
32			Interrupt structure of 8086,vector interrupt table, interrupt service routine	Regular	OHP,BB	
34			Introduction to DOS and BIOS interrupts	Missing	BB	
35			Interrupt controller-8259, Interfacing DMA controller 8257 to 8086	Additional	BB	
36			Tutorial class	Regular	OHPBB	
37			8251 USART architecture	Regular	BB	
			8251 USART interfacing	Regular	BB	
38			RS-232,IEEE 488 Standards	Missing	OHP,BB	
39			Prototyping and troubleshooting	Missing	OHP,BB	
40			TTL to RS 232 and RS-232 to TTL conversion	Missing	BB	
41	IV	08	Overview of 8051 microcontrollers	Regular	OHP,BB	
42			Architecture of 8051 microcontroller	Regular	BB	
43			I/O ports, memory segmentation of 8051	Regular	BB	
44			Addressing modes of 8051	Regular	BB	
45			Instruction set of 8051	Regular	BB	
46			Simple programs on 8051	Regular	BB	
47			Simple programs on 8051	Regular	OHP,BB	
49			Tutorial class	Regular	OHP,BB	
50	V	10	Interrupts of 8051 microcontroller	Regular	OHP,BB	
51			Timer/counters of 8051	Regular	BB	
52			Serial communication of 8051	Regular	BB	
53			Programming timer interrupts	Regular	OHP,BB	
54			Programming external hardware interrupts	Regular	OHP,BB	
55			Programming the serial communication interrupts	Regular	BB	
56			Programming 8051 timers/counters	Regular	BB	
57			Tutorial class	Regular	BB	
58			Distinguish between RiSC vs CISC. AVR RISC microcontroller – architecture	Additional	OHP,BB	
59			Introduction to PIC and ARM controllers	Additional	OHP,BB	

Micro Plan:ECE-A

SL. No	Unit No.	Total No. of Periods	Date	Topic to be covered in One lecture	Regular/ Additional/ missing	Teaching aids LCD/OHP/ BB	Remarks
1	1	11		Overview of basics from Switching Theory, Logic Design and Computer Organization.	Additional	BB	
2				Overview of MPMC syllabus and its importance in Real world	Additional	BB	
3				Evolution of microprocessors and Introduction to 8085 microprocessor,	Additional	OHP,BB	
4				Architecture of 8086 Microprocessor – functional diagram description	Regular	OHP,BB	
5				Register organization, memory segmentation	Regular	OHP,BB	
6				Programming model, memory addresses	Regular	OHP,BB	
7				Physical memory organization	Regular	BB	
8				Signal description of 8086, Common function signals	Regular	BB	
9				maximum and minimum mode signals	Regular	BB	
10				Timing diagrams, Interrupts of 8086	Regular	OHP,BB	
11				Tutorial class	Regular	BB	
12	II	08		Instruction formats, addressing modes	Regular	OHP,BB	
13				Instruction set of 8086	Regular	BB	
14				Instruction set of 8086	Regular	BB	
15				Assembler directives, macros	Regular	OHP,BB	
16				Simple programs on logical ,branch and call instructions	Regular	BB	
17				Programs on sorting, string manipulations	Regular	BB	
18				Tutorial class	Regular	BB	
19				Introduction to Pentium and dual core processors	Additional	OHP,BB	
20	III	20		Introduction to parallel data transfer methods and importance of port structure	Regular	OHP,BB	
21				8255-PPI	Regular	OHP,BB	
22				Modes of operation and interfacing to 8086	Regular	BB	

23				Modes of operation and interfacing to 8086	Regular	BB	
24				Interfacing keyboard, display to 8086	Regular	OHP,BB	
25				Stepper motor interfacing to 8086	Missing	BB	
26				A/D converter interfacing to 8086	Regular	OHP,BB	
27				D/A converter interfacing to 8086	Regular	BB	
28				Tutorial class	Regular	LCD,OHP,BB	
29				Memory interfacing to 8086	Regular	OHP,BB	
30				Memory interfacing to 8086	Regular	BB	
32				Interrupt structure of 8086,vector interrupt table, interrupt service routine	Regular	OHP,BB	
34				Introduction to DOS and BIOS interrupts	Missing	BB	
35				Interrupt controller-8259, Interfacing DMA controller 8257 to 8086	Additional	BB	
36				Tutorial class	Regular	OHPBB	
37				8251 USART architecture	Regular	BB	
				8251 USART interfacing	Regular	BB	
38				RS-232,IEEE 488 Standards	Missing	OHP,BB	
39				Prototyping and troubleshooting	Missing	OHP,BB	
40				TTL to RS 232 and RS-232 to TTL conversion	Missing	BB	
41	IV	08		Overview of 8051 microcontrollers	Regular	OHP,BB	
42				Architecture of 8051microcontroller	Regular	BB	
43				I/O ports, memory segmentation of 8051	Regular	BB	
44				Addressing modes of 8051	Regular	BB	
45				Instruction set of 8051	Regular	BB	
46				Simple programs on 8051	Regular	BB	
47				Simple programs on 8051	Regular	OHP,BB	
49				Tutorial class	Regular	OHP,BB	
50	V	10		Interrupts of 8051 microcontroller	Regular	OHP,BB	

51				Timer/counters of 8051	Regular	BB	
52				Serial communication of 8051	Regular	BB	
53				Programming timer interrupts	Regular	OHP,BB	
54				Programming external hardware interrupts	Regular	OHP,BB	
55				Programming the serial communication interrupts	Regular	BB	
56				Programming 8051 timers/counters	Regular	BB	

ECE-B

SL. No	Unit No.	Total No. of Periods		Topic to be covered in One lecture	Regular/ Additional/ missing	Teaching aids LCD/OHP/ BB	Remarks
1	1	11		Overview of basics from Switching Theory, Logic Design and Computer Organization.	Additional	BB	
2				Overview of MPMC syllabus and its importance in Real world	Additional	BB	
3				Evolution of microprocessors and Introduction to 8085 microprocessor,	Additional	OHP,BB	
4				Architecture of 8086 Microprocessor – functional diagram description	Regular	OHP,BB	
5				Register organization, memory segmentation	Regular	OHP,BB	
6				Programming model, memory addresses	Regular	OHP,BB	
7				Physical memory organization	Regular	BB	
8				Signal description of 8086, Common function signals	Regular	BB	
9				maximum and minimum mode signals	Regular	BB	
10				Timing diagrams, Interrupts of 8086	Regular	OHP,BB	
11				Tutorial class	Regular	BB	
12	II	08		Instruction formats, addressing modes	Regular	OHP,BB	
13				Instruction set of 8086	Regular	BB	
14				Instruction set of 8086	Regular	BB	

15				Assembler directives, macros	Regular	OHP,BB	
16				Simple programs on logical ,branch and call instructions	Regular	BB	
17				Programs on sorting, string manipulations	Regular	BB	
18				Tutorial class	Regular	BB	
19				Introduction to Pentium and dual core processors	Additional	OHP,BB	
20	III	20		Introduction to parallel data transfer methods and importance of port structure	Regular	OHP,BB	
21				8255-PPI	Regular	OHP,BB	
22				Modes of operation and interfacing to 8086	Regular	BB	
23				Modes of operation and interfacing to 8086	Regular	BB	
24				Interfacing keyboard, display to 8086	Regular	OHP,BB	
25				Stepper motor interfacing to 8086	Missing	BB	
26				A/D converter interfacing to 8086	Regular	OHP,BB	
27				D/A converter interfacing to 8086	Regular	BB	
28				Tutorial class	Regular	LCD,OHP,I	
29				Memory interfacing to 8086	Regular	OHP,BB	
30				Memory interfacing to 8086	Regular	BB	
32				Interrupt structure of 8086,vector interrupt table, interrupt service routine	Regular	OHP,BB	
34				Introduction to DOS and BIOS interrupts	Missing	BB	
35				Interrupt controller-8259, Interfacing DMA controller 8257 to 8086	Additional	BB	
36				Tutorial class	Regular	OHPBB	
37				8251 USART architecture	Regular	BB	
				8251 USART interfacing	Regular	BB	
38				RS-232,IEEE 488 Standards	Missing	OHP,BB	
39				Prototyping and troubleshooting	Missing	OHP,BB	
40				TTL to RS 232 and RS-232 to TTL conversion	Missing	BB	
41	IV	08		Overview of 8051 microcontrollers	Regular	OHP,BB	

42				Architecture of 8051 microcontroller	Regular	BB	
43				I/O ports, memory segmentation of 8051	Regular	BB	
44				Addressing modes of 8051	Regular	BB	
45				Instruction set of 8051	Regular	BB	
46				Simple programs on 8051	Regular	BB	
47				Simple programs on 8051	Regular	OHP, BB	
49				Tutorial class	Regular	OHP, BB	
50	V	10		Interrupts of 8051 microcontroller	Regular	OHP, BB	
51				Timer/counters of 8051	Regular	BB	
52				Serial communication of 8051	Regular	BB	
53				Programming timer interrupts	Regular	OHP, BB	
54				Programming external hardware interrupts	Regular	OHP, BB	
55				Programming the serial communication interrupts	Regular	BB	
56				Programming 8051 timers/counters	Regular	BB	

ECE-C

SL. No	Unit No.	Total No. of Periods	Date	Topic to be covered in One lecture	Regular/ Additional/ missing	Teaching aids LCD/OHP/ BB	Remarks
1	1	11		Overview of basics from Switching Theory, Logic Design and Computer Organization.	Additional	BB	
2				Overview of MPMC syllabus and its importance in Real world	Additional	BB	
3				Evolution of microprocessors and Introduction to 8085 microprocessor,	Additional	OHP, BB	
4				Architecture of 8086 Microprocessor – functional diagram description	Regular	OHP, BB	
5				Register organization, memory segmentation	Regular	OHP, BB	
6				Programming model, memory addresses	Regular	OHP, BB	
7				Physical memory organization	Regular	BB	

8				Signal description of 8086, Common function signals	Regular	BB	
9				maximum and minimum mode signals	Regular	BB	
10				Timing diagrams, Interrupts of 8086	Regular	OHP,BB	
11				Tutorial class	Regular	BB	
12	II	08		Instruction formats, addressing modes	Regular	OHP,BB	
13				Instruction set of 8086	Regular	BB	
14				Instruction set of 8086	Regular	BB	
15				Assembler directives, macros	Regular	OHP,BB	
16				Simple programs on logical ,branch and call instructions	Regular	BB	
17				Programs on sorting, string manipulations	Regular	BB	
18				Tutorial class	Regular	BB	
19				Introduction to Pentium and dual core processors	Additional	OHP,BB	
20	III	20		Introduction to parallel data transfer methods and importance of port structure	Regular	OHP,BB	
21				8255-PPI	Regular	OHP,BB	
22				Modes of operation and interfacing to 8086	Regular	BB	
23				Modes of operation and interfacing to 8086	Regular	BB	
24				Interfacing keyboard, display to 8086	Regular	OHP,BB	
25				Stepper motor interfacing to 8086	Missing	BB	
26				A/D converter interfacing to 8086	Regular	OHP,BB	
27				D/A converter interfacing to 8086	Regular	BB	
28				Tutorial class	Regular	LCD,OHP,BB	
29				Memory interfacing to 8086	Regular	OHP,BB	
30				Memory interfacing to 8086	Regular	BB	
32				Interrupt structure of 8086,vector interrupt table, interrupt service routine	Regular	OHP,BB	
34				Introduction to DOS and BIOS interrupts	Missing	BB	
35				Interrupt controller-8259, Interfacing DMA controller 8257 to 8086	Additional	BB	

36				Tutorial class	Regular	OHPBB	
37				8251 USART architecture	Regular	BB	
				8251 USART interfacing	Regular	BB	
38				RS-232,IEEE 488 Standards	Missing	OHP,BB	
39				Prototyping and troubleshooting	Missing	OHP,BB	
40				TTL to RS 232 and RS-232 to TTL conversion	Missing	BB	
41	IV	08		Overview of 8051 microcontrollers	Regular	OHP,BB	
42				Architecture of 8051microcontroller	Regular	BB	
43				I/O ports, memory segmentation of 8051	Regular	BB	
44				Addressing modes of 8051	Regular	BB	
45				Instruction set of 8051	Regular	BB	
46				Simple programs on 8051	Regular	BB	
47				Simple programs on 8051	Regular	OHP,BB	
49				Tutorial class	Regular	OHP,BB	
50	V	10		Interrupts of 8051 microcontroller	Regular	OHP,BB	
51				Timer/counters of 8051	Regular	BB	
52				Serial communication of 8051	Regular	BB	
53				Programming timer interrupts	Regular	OHP,BB	
54				Programming external hardware interrupts	Regular	OHP,BB	
55				Programming the serial communication interrupts	Regular	BB	
56				Programming 8051 timers/counters	Regular	BB	

ECE –D

SL. No	Unit No.	Total No. of	Date	Topic to be covered in One lecture	Regular/ Additional/	Teaching aids	Remarks
--------	----------	--------------	------	------------------------------------	----------------------	---------------	---------

		Periods			missing	LCD/OHP/BB	
1	I	11		Overview of basics from Switching The Logic Design and Computer Organization.	Additional	BB	
2				Overview of MPMC syllabus and its importance in Real world	Additional	BB	
3				Evolution of microprocessors and Introduction to 8085 microprocessor,	Additional	OHP, BB	
4				Architecture of 8086 Microprocessor – functional diagram description	Regular	OHP, BB	
5				Register organization, memory segmentation	Regular	OHP, BB	
6				Programming model, memory addresses	Regular	OHP, BB	
7				Physical memory organization	Regular	BB	
8				Signal description of 8086, Common function signals	Regular	BB	
9				maximum and minimum mode signals	Regular	BB	
10				Timing diagrams, Interrupts of 8086	Regular	OHP, BB	
11				Tutorial class	Regular	BB	
12	II	08		Instruction formats, addressing modes	Regular	OHP, BB	
13				Instruction set of 8086	Regular	BB	
14				Instruction set of 8086	Regular	BB	
15				Assembler directives, macros	Regular	OHP, BB	
16				Simple programs on logical ,branch and call instructions	Regular	BB	
17				Programs on sorting, string manipulations	Regular	BB	
18				Tutorial class	Regular	BB	
19				Introduction to Pentium and dual core processors	Additional	OHP, BB	
20	III	20		Introduction to parallel data transfer methods and importance of port structure	Regular	OHP, BB	
21				8255-PPI	Regular	OHP, BB	
22				Modes of operation and interfacing to 8086	Regular	BB	
23				Modes of operation and interfacing to 8086	Regular	BB	
24				Interfacing keyboard, display to 8086	Regular	OHP, BB	
25				Stepper motor interfacing to 8086	Missing	BB	

26				A/D converter interfacing to 8086	Regular	OHP,BB	
27				D/A converter interfacing to 8086	Regular	BB	
28				Tutorial class	Regular	LCD,OHP,BB	
29				Memory interfacing to 8086	Regular	OHP,BB	
30				Memory interfacing to 8086	Regular	BB	
32				Interrupt structure of 8086,vector interrupt table, interrupt service routine	Regular	OHP,BB	
34				Introduction to DOS and BIOS interrupts	Missing	BB	
35				Interrupt controller-8259, Interfacing DMA controller 8257 to 8086	Additional	BB	
36				Tutorial class	Regular	OHPBB	
37				8251 USART architecture	Regular	BB	
				8251 USART interfacing	Regular	BB	
38				RS-232,IEEE 488 Standards	Missing	OHP,BB	
39				Prototyping and troubleshooting	Missing	OHP,BB	
40				TTL to RS 232 and RS-232 to TTL conversion	Missing	BB	
41	IV	08		Overview of 8051 microcontrollers	Regular	OHP,BB	
42				Architecture of 8051microcontroller	Regular	BB	
43				I/O ports, memory segmentation of 8051	Regular	BB	
44				Addressing modes of 8051	Regular	BB	
45				Instruction set of 8051	Regular	BB	
46				Simple programs on 8051	Regular	BB	
47				Simple programs on 8051	Regular	OHP,BB	
49				Tutorial class	Regular	OHP,BB	
50	V	10		Interrupts of 8051 microcontroller	Regular	OHP,BB	
51				Timer/counters of 8051	Regular	BB	
52				Serial communication of 8051	Regular	BB	
53				Programming timer interrupts	Regular	OHP,BB	

54				Programming external hardware interrupts	Regular	OHP,BB	
55				Programming the serial communication interrupts	Regular	BB	
56				Programming 8051 timers/counters	Regular	BB	

14.Detailed notes

UNIT-I

8085 Microprocessor *Contents*

- General definitions
- Overview of 8085 microprocessor
- Overview of 8086 microprocessor
- Signals and pins of 8086 microprocessor

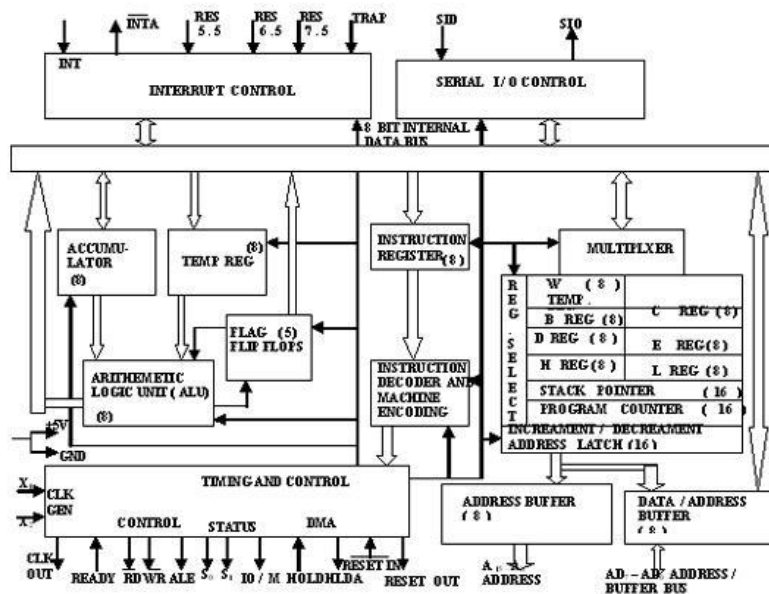
The salient features of 8085 μ p are:

- It is a 8 bit microprocessor.
- It is manufactured with N-MOS technology.
- It has 16-bit address bus and hence can address up to 2^{16} =65536 bytes (64KB) memory locations through A_0 - A_{15} .
- The first 8 lines of address bus and 8 lines of data bus are multiplexed AD_0 – AD_7 .
- Data bus is a group of 8 lines D_0 – D_7 .
- It supports external interrupt request.
- A 16 bit program counter (PC)
- A 16 bit stack pointer (SP)
- Six 8-bit general purpose register arranged in pairs: BC, DE, HL.
- It requires a signal +5V power supply and operates at 3.2 MHZ single phase clock.
- It is enclosed with 40 pins DIP (Dual in line package).

Overview of 8085 microprocessor

8085 Architecture

- Pin Diagram
- Functional Block Diagram



Instruction Set

8085 instruction set consists of the following instructions:

- Data moving instructions.
- Arithmetic - add, subtract, increment and decrement.
- Logic - AND, OR, XOR and rotate.
- Control transfer - conditional, unconditional, call subroutine, return from subroutine and restarts.
- Input/Output instructions.
- Other - setting/clearing flag bits, enabling/disabling interrupts, stack operations, etc.

Addressing mode

• **Register** - references the data in a register or in a register pair.

Register indirect - instruction specifies register pair containing address, where the data is located.

Direct, Immediate - 8 or 16-bit data.

Advantages of 8086 over 8085 are:

1. pipelining is employed making the execution faster.
2. data bus width increased to 16 bits.

3. higher memory of 1MB.

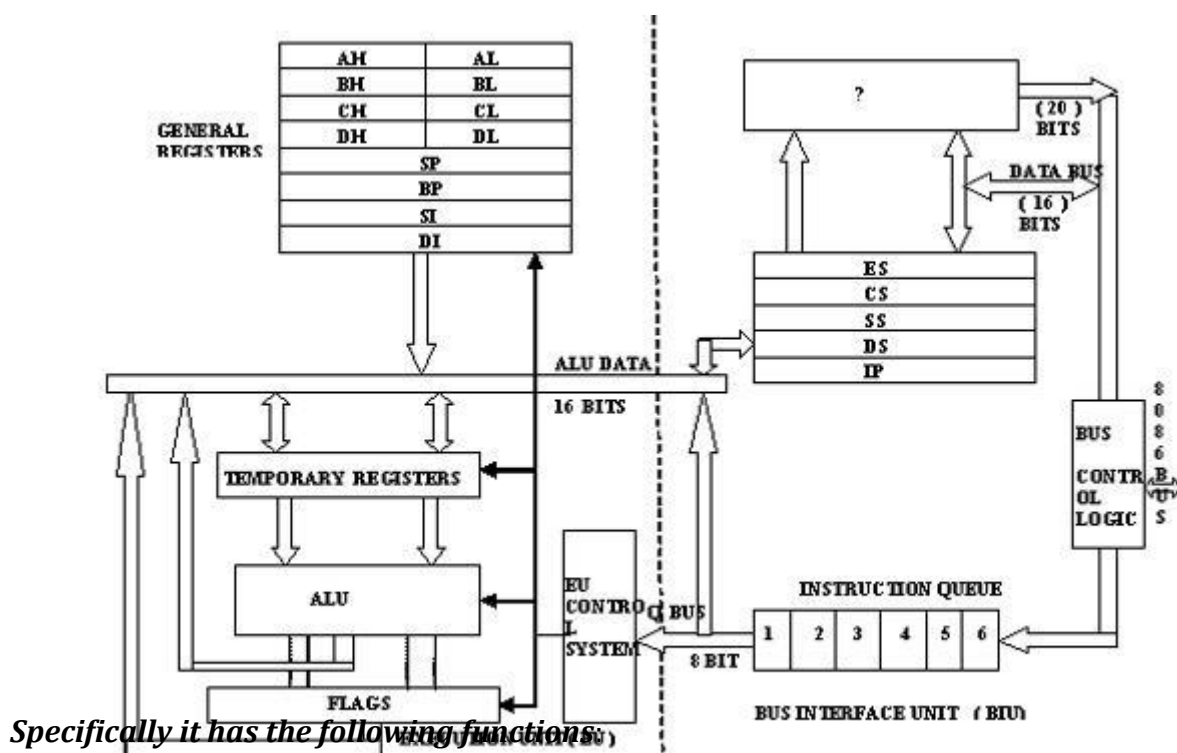
4. some instructions such as MUL or DIV are available for multiplication and division.

5. increased instruction set making the programming easier.

- EU executes instructions from the instruction system byte queue.
- Both units operate asynchronously to give the 8086 an overlapping instruction fetch and execution mechanism which is called as Pipelining. This results in efficient use of the system bus and system performance.
- BIU contains Instruction queue, Segment registers, Instruction pointer, Address adder.
- EU contains Control circuitry, Instruction decoder, ALU, Pointer and Index register, Flag register.

BUS INTERFACR UNIT:

- It provides a full 16 bit bidirectional data bus and 20 bit address bus.
- The bus interface unit is responsible for performing all external bus operations.



Specifically it has the following functions,

- Instruction fetch, Instruction queuing, Operand fetch and storage, Address relocation and Bus control.

•The BIU uses a mechanism known as an instruction stream queue to implement a pipeline architecture.

- This queue permits prefetch of up to six bytes of instruction code. When ever the queue of the BIU is not full, it has room for at least two more bytes and at the same time the EU is not requesting it to read or write operands from memory, the BIU is free to look ahead in the program by prefetching the next sequential instruction.
- These prefetching instructions are held in its FIFO queue. With its 16 bit data bus, the BIU fetches two instruction bytes in a single memory cycle.
- After a byte is loaded at the input end of the queue, it automatically shifts up through the FIFO to the empty location nearest the output.
- The EU accesses the queue from the output end. It reads one instruction byte after the other from the output of the queue. If the queue is full and the EU is not requesting access to operand in memory.
- These intervals of no bus activity, which may occur between bus cycles are known as ***Idle state***.
- If the BIU is already in the process of fetching an instruction when the EU request it to read or write operands from memory or I/O, the BIU first completes the instruction fetch bus cycle before initiating the operand read / write cycle.
- The BIU also contains a dedicated adder which is used to generate the 20bit physical address that is output on the address bus. This address is formed by combining the current contents of the code segment CS register and the current contents of the instruction pointer IP register.
- The BIU is also responsible for generating bus control signals such as those for memory read or write and I/O read or write.

EXECUTION UNIT

The Execution unit is responsible for decoding and executing all instructions.

- The EU extracts instructions from the top of the queue in the BIU, decodes them, generates operands if necessary, passes them to the BIU and requests it to perform the read or write bus cycles to memory or I/O and perform the operation specified by the instruction on the operands.
- During the execution of the instruction, the EU tests the status and control flags and updates them based on the results of executing the instruction.
- If the queue is empty, the EU waits for the next instruction byte to be fetched and shifted to top of the queue.
- When the EU executes a branch or jump instruction, it transfers control to a location

corresponding to another set of sequential instructions.

SPECIAL FUNCTIONS OF GENERAL PURPOSE REGISTERS

Accumulator register consists of 2 8-bit registers AL and AH, which can be combined together and used as a 16-bit register AX. AL in this case contains the low-order byte of the word, and AH contains the high-order byte. Accumulator can be used for I/O operations and string manipulation.

Base register consists of 2 8-bit registers BL and BH, which can be combined together and used as a 16-bit register BX. BL in this case contains the low-order byte of the word, and BH contains the high-order byte. BX register usually contains a data pointer used for based, based indexed or register indirect addressing.

Count register consists of 2 8-bit registers CL and CH, which can be combined together and used as a 16-bit register CX. When combined, CL register contains the low-order byte of the word, and CH contains the high-order byte. Count register can be used as a counter in string manipulation and shift/rotate instructions.

Data register consists of 2 8-bit registers DL and DH, which can be combined together and used as a 16-bit register DX. When combined, DL register contains the low-order byte of the word, and DH contains the high-order byte. Data register can be used as a port number in I/O operations. In integer 32-bit multiply and divide instruction the DX register contains high-order word of the initial or resulting number.

SPECIAL FUNCTIONS OF SPECIAL PURPOSE REGISTERS

Stack Pointer (SP) is a 16-bit register pointing to program stack.

Base Pointer (BP) is a 16-bit register pointing to data in stack segment. BP register is usually used for based, based indexed or register indirect addressing.

Source Index (SI) is a 16-bit register. SI is used for indexed, based indexed and register indirect addressing, as well as a source data address in string manipulation instructions.

Destination Index (DI) is a 16-bit register. DI is used for indexed, based indexed and register indirect addressing, as well as a destination data address in string manipulation instructions. The si and di registers (Source Index and Destination Index) have some special purposes as well. You may use these registers as pointers (much like the bx register) to indirectly access memory. You'll also use these registers with the 8086 string instructions when processing character strings.

The bp register (Base Pointer) is similar to the bx register. You'll generally use this register to access parameters and local variables in a procedure.

The sp register (Stack Pointer) has a very special purpose - it maintains the *program stack*. Normally, you would not use this register for arithmetic computations. The proper operation of most programs depends upon the careful use of this register.

SEGMENTATION:

Since address registers and address operands are only 16 bits they can only address 64k bytes. In order to address the 20-bit address range of the 8086, physical addresses (those that are put on the address bus) are always formed by adding the values of one of the instruction is executed? The use of segment registers reduces the size of pointers to 16 bits.

This reduces the code size but also restricts the addressing range of a pointer to 64k bytes. Performing address arithmetic within data structures larger than 64k is awkward. This is the biggest drawback of the 8086 architecture. We will restrict ourselves to short programs where all of the code, data and stack are placed into the same 64k segment (i.e. CS=DS=SS).

Most of the registers contain data/instruction offsets within 64 KB memory segment. There are four different 64 KB segments for instructions, stack, data and extra data. To specify where in 1 MB of processor memory these 4 segments are located the processor uses four segment registers:

Memory

- Program, data and stack memories occupy the same memory space. As the most of the processor instructions use 16-bit pointers the processor can effectively address only 64 KB of memory.

- To access memory outside of 64 KB the CPU uses special segment registers to specify where the code, stack and data 64 KB segments are positioned within 1 MB of memory (see the "Registers" section below).

- 16-bit pointers and data are stored as:

address: low-order byte

address+1: high-order byte

- **Program memory** - program can be located anywhere in memory. Jump and call instructions can be used for short jumps within currently selected 64 KB code segment, as well as for far jumps anywhere within 1 MB of memory.

- All conditional jump instructions can be used to jump within approximately +127 to -127 bytes from current instruction.

- Data memory** - the processor can access data in any one out of 4 available segments, which limits the size of accessible memory to 256 KB (if all four segments point to different 64 KB blocks).

- Accessing data from the Data, Code, Stack or Extra segments can be usually done by prefixing instructions with the DS:, CS:, SS: or ES: (some registers and instructions by default may use the ES or SS segments instead of DS segment).

- Word data can be located at odd or even byte boundaries. The processor uses two memory accesses to read 16-bit word located at odd byte boundaries. Reading word data from even byte boundaries requires only one memory access.

- Stack memory** can be placed anywhere in memory. The stack can be located at odd memory addresses, but it is not recommended for performance reasons (see "Data Memory" above).

Reserved locations:

- 0000h - 03FFh are reserved for interrupt vectors. Each interrupt vector is a 32-bit pointer in format segment: offset.

- FFFF0h - FFFFFh - after RESET the processor always starts program execution at the FFFF0h address.

segment registers to the 16-bit address to form a 20-bit address.

The segment registers themselves only contain the most-significant 16 bits of the 20-bit value that is contributed by the segment registers. The least significant four bits of the segment address are always zero.

By default, the DS (data segment) is used for data transfer instructions (e.g. MOV), CS (code segment) is used with control transfer instructions (e.g. JMP or CALL), and SS is used with the stack pointer (e.g. PUSH or to save/restore addresses during CALL/RET or INT instructions).

Exercise: If DS contains 0100H, what address will be written by the instruction MOV [2000H], AL? If CX contains 1122H, SP contains 1234H, and SS contains 2000H, what memory values will change and what will be their values when the PUSH CX

Code segment (CS) is a 16-bit register containing address of 64 KB segment with processor instructions. The processor uses CS segment for all accesses to instructions referenced by instruction pointer (IP) register. CS register cannot be changed directly.

The CS register is automatically updated during far jump, far call and far return instructions.

Stack segment (SS) is a 16-bit register containing address of 64KB segment with program stack. By default, the processor assumes that all data referenced by the stack pointer (SP) and base pointer (BP) registers is located in the stack segment. SS register can be changed directly using POP instruction.

Data segment (DS) is a 16-bit register containing address of 64KB segment with program data. By default, the processor assumes that all data referenced by general registers (AX, BX, CX, DX) and index register (SI, DI) is located in the data segment. DS register can be changed directly using POP and LDS instructions.

Extra segment (ES) is a 16-bit register containing address of 64KB segment, usually with program data. By default, the processor assumes that the DI register references the ES segment in string manipulation instructions. ES register can be changed directly using POP and LES instructions.

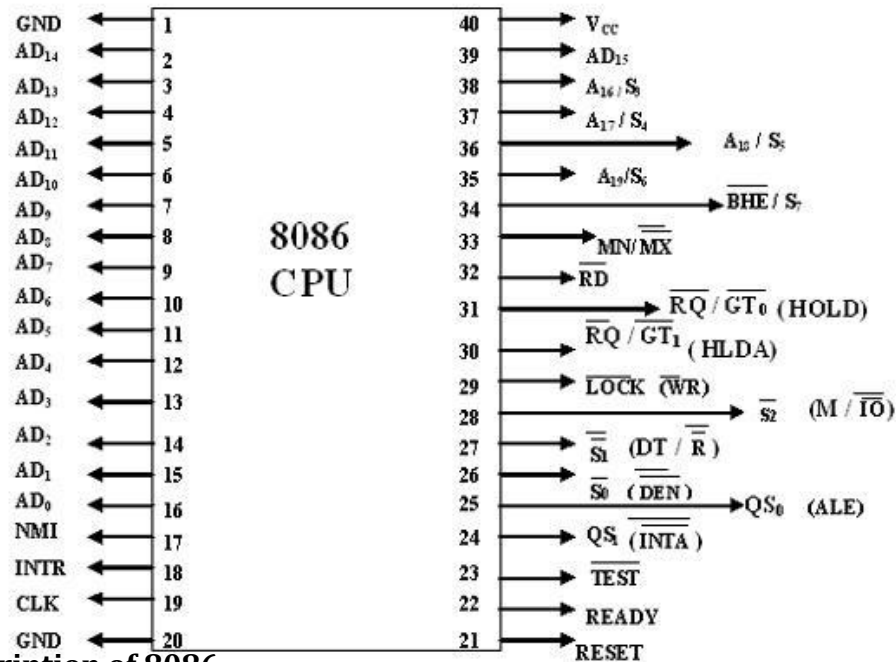
It is possible to change default segments used by general and index registers by prefixing instructions with a CS, SS, DS or ES prefix.

8086 FLAG REGISTER

Flags is a 16-bit register containing 9 1-bit flags:

- Overflow Flag (OF) - set if the result is too large positive number, or is too small negative number to fit into destination operand.
- Direction Flag (DF) - if set then string manipulation instructions will auto-decrement index registers. If cleared then the index registers will be auto-incremented.
- Interrupt-enable Flag (IF) - setting this bit enables maskable interrupts.
- Single-step Flag (TF) - if set then single-step interrupt will occur after the next instruction.
- Sign Flag (SF) - set if the most significant bit of the result is set.
- Zero Flag (ZF) - set if the result is zero.
- Auxiliary carry Flag (AF) - set if there was a carry from or borrow to bits 0-3 in the AL register.
- Parity Flag (PF) - set if parity (the number of "1" bits) in the low-order byte of the result is even.

Carry Flag (CF) - set if there was a carry from or borrow to the most significant bit during last result calculation



Signal Description of 8086

The Microprocessor 8086 is a 16-bit CPU available in different clock rates and packaged in a 40 pin Cerdip or plastic package.

- The 8086 operates in single processor or multiprocessor configuration to achieve high performance. The pins serve a particular function in minimum mode (single processor mode) and other function in maximum mode configuration (multiprocessor mode).
- The 8086 signals can be categorized in three groups. The first are the signal having common functions in minimum as well as maximum mode
- The second are the signals which have special functions for minimum mode and third are the signals having special functions for maximum mode.

The following signal descriptions are common for both modes.

AD15-AD0 : These are the time multiplexed memory I/O address and data lines.

- Address remains on the lines during T1 state, while the data is available on the data bus during T2, T3, Tw and T4.
- These lines are active high and float to a tristate during interrupt acknowledge and local bus hold acknowledge cycles

A19/S6,A18/S5,A17/S4,A16/S3 : These are the time multiplexed address and status lines.

- During T1 these are the most significant address lines for memory operations.

- During I/O operations, these lines are low. During memory or I/O operations, status information is available on those lines for T2, T3, Tw and T4.
- The status of the interrupt enable flag bit is updated at the beginning of each clock cycle.
- The S4 and S3 combinedly indicate which segment register is presently being used for memory accesses as in below fig.
- These lines float to tri-state off during the local bus hold acknowledge. The status line S6 is always low .
- The address bit are separated from the status bit using latches controlled by the ALE signal.

BHE /S7 : The bus high enable is used to indicate the transfer of data over the higherorder (D15-D8) data bus as shown in table. It goes low for the data transfer over D15- D8 and is used to derive chip selects of odd address memory bank or peripherals. BHE is low during T1 for read, write and interrupt acknowledge cycles, whenever a byte is to be transferred on higher byte of data bus. The status information is available during T2, T3 and T4. The signal is active low and tristated during hold. It is low during T1 for the first pulse of the interrupt acknowledge cycle.

BHE	A₀	Indication
0	0	Whole word
0	1	Upper byte from or to even address
1	0	Lower byte from or to even address
1	1	None

RD – Read : This signal on low indicates the peripheral that the processor is performing s memory or I/O read operation. RD is active low and shows the state for T2, T3, Tw of any read cycle. The signal remains tristated during the hold acknowledge.

• **READY** : This is the acknowledgement from the slow device or memory that they have completed the data transfer. The signal made available by the devices is synchronized by the 8284A clock generator to provide ready input to the 8086. the signal is active high.

• **INTR-Interrupt Request** : This is a triggered input. This is sampled during the last clock cycles of each instruction to determine the availability of the request. If any interrupt request is pending, the processor enters the interrupt acknowledge cycle.

• This can be internally masked by resulting the interrupt enable flag. This signal is

active high and internally synchronized.

- **TEST** : This input is examined by a 'WAIT' instruction. If the TEST pin goes low, execution will continue, else the processor remains in an idle state. The input is synchronized internally during each clock cycle on leading edge of clock.

- **CLK**- Clock Input : The clock input provides the basic timing for processor operation and bus control activity. Its an asymmetric square wave with 33% duty cycle

MN/ MX : The logic level at this pin decides whether the processor is to operate in either minimum or maximum mode.

- **The following pin functions are for the minimum mode operation of 8086.**

- **M/ IO – Memory/IO** : This is a status line logically equivalent to S2 in maximum mode. When it is low, it indicates the CPU is having an I/O operation, and when it is high, it indicates that the CPU is having a memory operation. This line Becomes active high in the previous T4 and remains active till final T4 of the current cycle. It is tri stated during local bus "hold acknowledge ".

INTA – Interrupt Acknowledge : This signal is used as a read strobe for interrupt acknowledge cycles. i.e. when it goes low, the processor has accepted the interrupt.

ALE – Address Latch Enable :This output signal indicates the availability of the valid address on the address/data lines, and is connected to latch enable input of latches. This signal is active high and is never tri stated.

- **DT/ R – Data Transmit/Receive**: This output is used to decide the direction of data flow through the transceivers (bidirectional buffers). When the processor sends out data, this signal is high and when the processor is receiving data, this signal is low.

- **DEN – Data Enable** :This signal indicates the availability of valid data over the address/data lines. It is used to enable the transceivers (bi directional buffers) to separate the data from the multiplexed address/data signal. It is active from the middle of T2 until the middle of T4. This is tri stated during hold acknowledge' cycle.

HOLD, HLDA- Acknowledge : When the HOLD line goes high, it indicates to the processor that another master is requesting the bus access. •The processor, after receiving the HOLD request, issues the hold acknowledge signal on HLDA pin, in the

middle of the next clock cycle after completing the current bus cycle. •At the same time, the processor floats the local bus and control lines. When the processor detects the HOLD line low, it lowers the HLDA signal. HOLD is an asynchronous input, and is should be externally synchronized. •If the DMA request is made while the CPU is performing a memory or I/O cycle, it will release the local bus during T4 provided :

- 1.The request occurs on or before T2 state of the current cycle.
- 2.The current cycle is not operating over the lower byte of a word.
- 3.The current cycle is not the first acknowledge of an interrupt acknowledge sequence.
4. A Lock instruction is not being executed

The following pin function are applicable for maximum mode operation of 8086.

•**S₂, S₁, S₀ – Status Lines** : These are the status lines which reflect the type of operation, being carried out by the processor. These become activity during T4 of the previous cycle and active during T1 and T2 of the current bus cycles.

S ₂	S ₁	S ₀	Indication
0	0	0	Interrupt Acknowledge
0	0	1	Read I/O port
0	1	0	Write I/O port
0	1	1	Halt
1	0	0	Code Access
1	0	1	Read memory
1	1	0	Write memory
1	1	1	Passive

LOCK: This output pin indicates that other system bus master will be prevented from gaining the system bus, while the LOCK signal is low

•The LOCK signal is activated by the 'LOCK' prefix instruction and remains active until the completion of the next instruction. When the CPU is executing a critical instruction which requires the system bus, the LOCK prefix instruction ensures that other processors connected in the system will not gain the control of the bus.

• The 8086, while executing the prefixed instruction, asserts the bus lock signal output, which may be connected to an external bus controller.

•**QS₁, QS₀ – Queue Status:** These lines give information about the status of the code-prefetch queue. These are active during the CLK cycle after while the queue operation is

performed.

- This modification in a simple fetch and execute architecture of a conventional microprocessor offers an added advantage of pipelined processing of the instructions.
- The 8086 architecture has 6-byte instruction prefetch queue. Thus even the largest (6-bytes) instruction can be prefetched from the memory and stored in the prefetch. This results in a faster execution of the instructions.
- In 8085 an instruction is fetched, decoded and executed and only after the execution of this instruction, the next one is fetched.
- By prefetching the instruction, there is a considerable speeding up in instruction execution in 8086. This is known as ***instruction pipelining***.
- At the starting the CS:IP is loaded with the required address from which the execution is to be started. Initially, the queue will be empty and the microprocessor starts a fetch operation to bring one byte (the first byte) of instruction code, if the CS:IP address is odd or two bytes at a time, if the CS:IP address is even.
- The first byte is a complete opcode in case of some instruction (one byte opcode instruction) and is a part of opcode, in case of some instructions (two byte opcode instructions), the remaining part of code lies in second byte.
- The second byte is then decoded in continuation with the first byte to decide the instruction length and the number of subsequent bytes to be treated as instruction data.
- The queue is updated after every byte is read from the queue but the fetch cycle is initiated by BIU only if at least two bytes of the queue are empty and the EU may be concurrently executing the fetched instructions
- **RQ / GT0 , RQ / GT1 – Request/Grant** : These pins are used by the other local bus master in maximum mode, to force the processor to release the local bus at the end of the processor current bus cycle.
- Each of the pin is bidirectional with RQ/GT0 having higher priority than RQ/GT1.
- RQ/GT pins have internal pull-up resistors and may be left unconnected.

Request/Grant sequence is as follows:

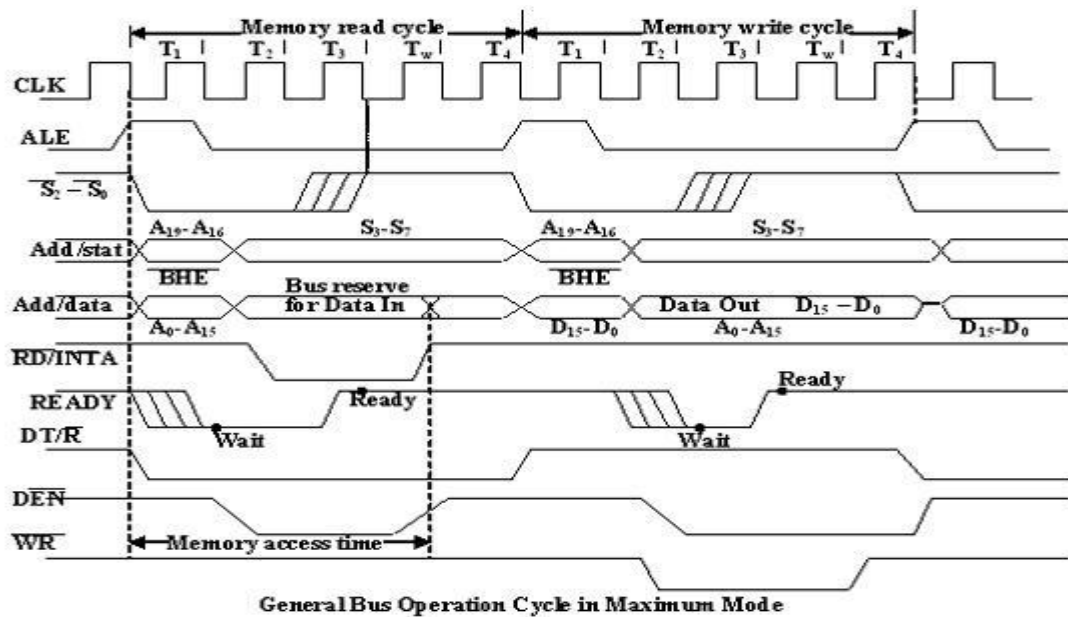
1. A pulse of one clock wide from another bus master requests the bus access to 8086.
2. During T4(current) or T1(next) clock cycle, a pulse one clock wide from 8086 to the requesting master, indicates that the 8086 has allowed the local bus to float and that it will enter the 'hold acknowledge' state at next cycle. The CPU bus interface unit is likely to be disconnected from the local bus of the system.

3. A one clock wide pulse from the another master indicates to the 8086 that the hold request is about to end and the 8086 may regain control of the local bus at the next clock cycle. Thus each master to master exchange of the local bus is a sequence of 3 pulses. There must be at least one dead clock cycle after each bus exchange.

- The request and grant pulses are active low.
- For the bus request those are received while 8086 is performing memory or I/O cycle, the granting of the bus is governed by the rules as in case of HOLD and HLDA in minimum mode.

General Bus Operation:

- The 8086 has a combined address and data bus commonly referred as a time multiplexed address and data bus.
- The main reason behind multiplexing address and data over the same pins is the maximum utilization of processor pins and it facilitates the use of 40 pin standard DIP package. Maximum utilization of processor pins and it facilitates the use of 40 pin standard DIP package.
- The bus can be de multiplexed using a few latches and transceivers, when ever required.
- Basically, all the processor bus cycles consist of at least four clock cycles. These are referred to as T1, T2, T3, T4. The address is transmitted by the processor during T1. It is present on the bus only for one cycle.
- The negative edge of this ALE pulse is used to separate the address and the data or status information. In maximum mode, the status lines S0, S1 and S2 are used to indicate the type of operation.
- In a minimum mode 8086 system, the microprocessor 8086 is operated in minimum mode by strapping its MN/MX pin to logic 1.
- In this mode, all the control signals are given out by the microprocessor chip itself. There is a single microprocessor in the minimum mode system.
- The remaining components in the system are latches, transceivers, clock generator, memory and I/O devices. Some type of chip selection logic may be required for selecting memory or I/O devices, depending upon the address map of the system.
- In a minimum mode 8086 system, the microprocessor 8086 is operated in minimum mode by strapping its MN/MX pin to logic 1.



- In this mode, all the control signals are given out by the microprocessor chip itself. There is a single microprocessor in the minimum mode system.
- The remaining components in the system are latches, transreceivers, clock generator, memory and I/O devices. Some type of chip selection logic may be required for selecting memory or I/O devices, depending upon the address map of the system.
- Latches are generally buffered output D-type flip-flops like 74LS373 or 8282. They are used for separating the valid address from the multiplexed address/data signals and are controlled by the ALE signal generated by 8086.
- Transreceivers are the bidirectional buffers and some times they are called as data amplifiers. They are required to separate the valid data from the time multiplexed address/data signals.
- They are controlled by two signals namely, DEN and DT/R. The DEN signal indicates the direction of data, i.e. from or to the processor. The system contains memory for the monitor and users program storage.
- Usually, EPROM are used for monitor storage, while RAM for users program storage. A system may contain I/O devices.
- The working of the minimum mode configuration system can be better described in terms of the timing diagrams rather than qualitatively describing the operations.
- The opcode fetch and read cycles are similar. Hence the timing diagram can be categorized in two parts, the first is the timing diagram for read cycle and the second is the timing diagram for write cycle.
- The read cycle begins in T1 with the assertion of address latch enable (ALE) signal and

also M / IO signal. During the negative going edge of this signal, the valid address is latched on the local bus.

- The BHE and A0 signals address low, high or both bytes. From T1 to T4 , the M/IO signal indicates a memory or I/O operation.
- At T2, the address is removed from the local bus and is sent to the output. The bus is then tristated. The read (RD) control signal is also activated in T2.
- The read (RD) signal causes the address device to enable its data bus drivers. After RD goes low, the valid data is available on the data bus.
- The addressed device will drive the READY line high. When the processor returns the read signal to high level, the addressed device will again tristate its bus drivers.
- A write cycle also begins with the assertion of ALE and the emission of the address. The M/IO signal is again asserted to indicate a memory or I/O operation. In T2, after sending the address in T1, the processor sends the data to be written to the addressed location.
- The data remains on the bus until middle of T4 state. The WR becomes active at the beginning of T2 (unlike RD is somewhat delayed in T2 to provide time for floating).
- The BHE and A0 signals are used to select the proper byte or bytes of memory or I/O word to be read or write.
- The M/IO, RD and WR signals indicate the type of data transfer as specified in table below.

Hold Response sequence: The HOLD pin is checked at leading edge of each clock pulse. If it is received active by the processor before T4 of the previous cycle or during T1 state of the current cycle, the CPU activates HLDA in the next clock cycle and for succeeding bus cycles, the bus will be given to another requesting master.

- The control of the bus is not regained by the processor until the requesting master does not drop the HOLD pin low. When the request is dropped by the requesting master, the HLDA is dropped by the processor at the trailing edge of the next clock.

Maximum Mode 8086 System

- In the maximum mode, the 8086 is operated by strapping the MN/MX pin to ground.
- In this mode, the processor derives the status signal S2, S1, S0. Another chip called bus controller derives the control signal using this status information .
- In the maximum mode, there may be more than one microprocessor in the system configuration.
- The components in the system are same as in the minimum mode system.

- The basic function of the bus controller chip IC8288, is to derive control signals like RD and WR (for memory and I/O devices), DEN, DT/R, ALE etc. using the information by the processor on the status lines.

- The bus controller chip has input lines S2, S1, S0 and CLK. These inputs to 8288 are driven by CPU.

- It derives the outputs ALE, DEN, DT/R, MRDC, MWTC, AMWC, IORC, IOWC and AIOWC. The AEN, IOB and CEN pins are specially useful for multiprocessor systems.

- AEN and IOB are generally grounded. CEN pin is usually tied to +5V. The significance of the MCE/PDEN output depends upon the status of the IOB pin.

- If IOB is grounded, it acts as master cascade enable to control cascade 8259A, else it acts as peripheral data enable used in the multiple bus configurations.

- INTA pin used to issue two interrupt acknowledge pulses to the interrupt controller or to an interrupting device

IORC, IOWC are I/O read command and I/O write command signals respectively .These signals enable an IO interface to read or write the data from or to the address port.

- The MRDC, MWTC are memory read command and memory write command signals respectively and may be used as memory read or write signals.

- All these command signals instructs the memory to accept or send data from or to the bus.

- For both of these write command signals, the advanced signals namely AIOWC and AMWTC are available.

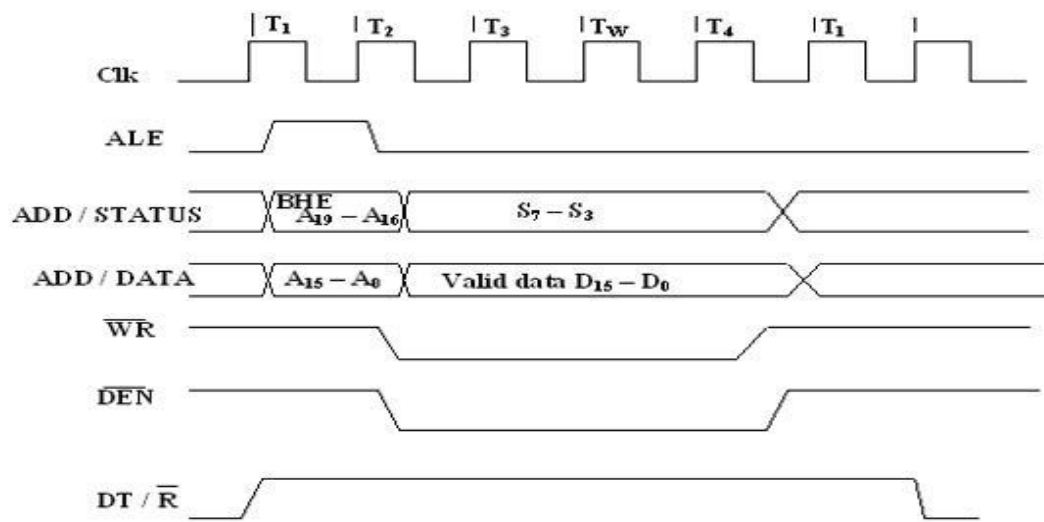
- Here the only difference between in timing diagram between minimum mode and maximum mode is the status signals used and the available control and advanced command signals.

- R0, S1, S2 are set at the beginning of bus cycle.8288 bus controller will output a pulse as on the ALE and apply a required signal to its DT / R pin during T1.

- In T2, 8288 will set DEN=1 thus enabling transceivers, and for an input it will activate MRDC or IORC. These signals are activated until T4. For an output, the AMWC or AIOWC is activated from T2 to T4 and MWTC or IOWC is activated from T3 to T4.

- The status bit S0 to S2 remains active until T3 and become passive during T3 and T4.

- If reader input is not activated before T3, wait state will be inserted between T3 and T4.



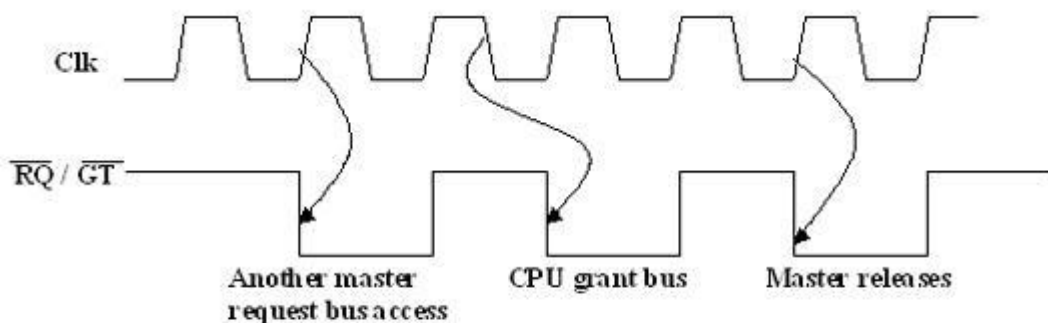
Write Cycle Timing Diagram for Minimum Mode

•Timings for RQ/ GT Signals :

The request/grant response sequence contains a series of three pulses. The request/grant pins are checked at each rising pulse of clock input.

When a request is detected and if the condition for HOLD request is satisfied, the processor issues a grant pulse over the RQ/GT pin immediately during T4 (current) or T1 (next) state.

When the requesting master receives this pulse, it accepts the control of the bus; it sends a release pulse to the processor using RQ/GT pin.



RQ/GT Timings in Maximum Mode.

Minimum Mode Interface

- When the Minimum mode operation is selected, the 8086 provides all control signals

needed to implement the memory and I/O interface. The minimum mode signal can be divided into the following basic groups : address/data bus, status, control, interrupt and DMA.

- **Address/Data Bus** : these lines serve two functions. As an address bus is 20 bits long and consists of signal lines A0 through A19. A19 represents the MSB and A0 LSB. A 20bit address gives the 8086 a 1Mbyte memory address space. More over it has an independent

I/O address space which is 64K bytes in length.

- The 16 data bus lines D0 through D15 are actually multiplexed with address lines A0 through A15 respectively. By multiplexed we mean that the bus work as an address bus during first machine cycle and as a data bus during next machine cycles. D15 is the MSB and D0 LSB.

- When acting as a data bus, they carry read/write data for memory, input/output data for

I/O devices, and interrupt type codes from an interrupt controller.

S ₄	S ₃	Segment Register
0	0	Extra
0	1	Stack
1	0	Code / none
1	1	Data

Memory segment status codes.

Status signal:

The four most significant address lines A19 through A16 are also multiplexed but in this case with status signals S6 through S3. These status bits are output on the bus at the same

time that data are transferred over the other bus lines.

- Bit S4 and S3 together form a 2 bit binary code that identifies which of the 8086

internal

segment registers are used to generate the physical address that was output on the address

bus during the current bus cycle.

- Code S4S3 = 00 identifies a register known as ***extra segment register*** as the source of the segment address.

- Status line S5 reflects the status of another internal characteristic of the 8086. It is the logic level of the internal enable flag. The last status bit S6 is always at the logic 0 level

•Control Signals :

The control signals are provided to support the 8086 memory I/O interfaces. They control functions such as when the bus is to carry a valid address in which direction data

are to be transferred over the bus, when valid write data are on the bus and when to put read data on the system bus.

ALE is a pulse to logic 1 that signals external circuitry when a valid address word is on the bus. This address must be latched in external circuitry on the 1-to-0 edge of the pulse

- Another control signal that is produced during the bus cycle is BHE bank high enable. Logic 0 on this used as a memory enable signal for the most significant byte half of the data bus D8 through D1. These lines also serves a second function, which is as the S₇ status line.

- Using the M/I/O and DT/R lines, the 8086 signals which type of bus cycle is in progress and in which direction data are to be transferred over the bus.

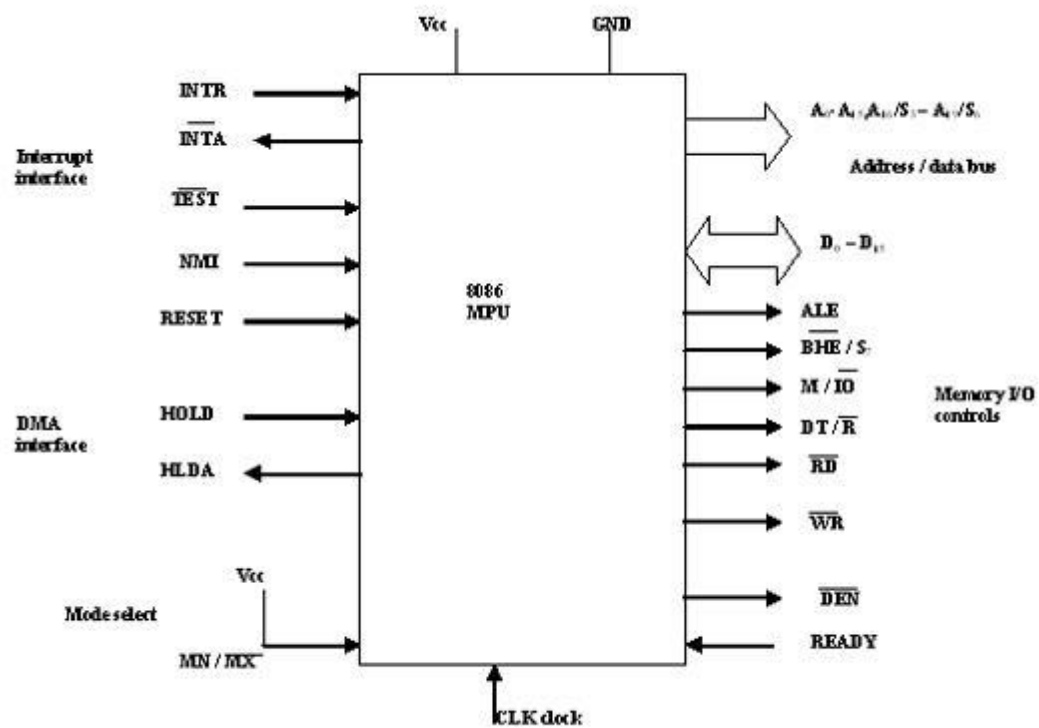
- The logic level of M/I/O tells external circuitry whether a memory or I/O transfer is taking place over the bus. Logic 1 at this output signals a memory operation and logic 0 an I/O operation.

- The direction of data transfer over the bus is signaled by the logic level output at DT/R. When this line is logic 1 during the data transfer part of a bus cycle, the bus is in the transmit mode. Therefore, data are either written into memory or output to an I/O device.

- On the other hand, logic 0 at DT/R signals that the bus is in the receive mode. This corresponds to reading data from memory or input of data from an input port.
- The signal read RD and write WR indicates that a read bus cycle or a write bus cycle is in progress. The 8086 switches WR to logic 0 to signal external device that valid write or output data are on the bus
- On the other hand, RD indicates that the 8086 is performing a read of data of the bus. During read operations, one other control signal is also supplied. This is DEN (data enable) and it signals external devices when they should put data on the bus.
- There is one other control signal that is involved with the memory and I/O interface. This is the READY signal.
- READY signal is used to insert wait states into the bus cycle such that it is extended by a number of clock periods. This signal is provided by an external clock generator device and can be supplied by the memory or I/O sub-system to signal the 8086 when they are ready to permit the data transfer to be completed

Maximum Mode Interface

- When the 8086 is set for the maximum-mode configuration, it provides signals for implementing a multiprocessor / coprocessor system environment.
- By multiprocessor environment we mean that one microprocessor exists in the system and that each processor is executing its own program.
- Usually in this type of system environment, there are some system resources that are common to all processors.
- They are called as ***global resources***. There are also other resources that are assigned to specific processors. These are known as ***local or private resources***.
- Coprocessor also means that there is a second processor in the system. In this two processor does not access the bus at the same time.
- One passes the control of the system bus to the other and then may suspend its operation.
- In the maximum-mode 8086 system, facilities are provided for implementing allocation of global resources and passing bus control to other microprocessor or coprocessor .



Block Diagram of the Minimum Mode 8086 MPU

UNIT-II

ADDRESSING MODES OF 8086:

Implied - the data value/data address is implicitly associated with the instruction.

Direct - the instruction operand specifies the memory address where data is located.

•**Register indirect** - instruction specifies a register containing an address, where data is located. This addressing mode works with SI, DI, BX and BP registers.

•**Register** - references the data in a register or in a register pair.

•**Immediate** - the data is provided in the instruction.

•**Based** :- 8-bit or 16-bit instruction operand is added to the contents of a base register (BX or BP), the resulting value is a pointer to location where data resides.

•**Indexed** :- 8-bit or 16-bit instruction operand is added to the contents of an index register (SI or DI), the resulting value is a pointer to location where data resides

•**Based Indexed** :- the contents of a base register (BX or BP) is added to the contents of an index register (SI or DI), the resulting value is a pointer to location where data resides.

•**Based Indexed with displacement** :- 8-bit or 16-bit instruction operand is added to the contents of a base register (BX or BP) and index register (SI or DI), the resulting value is a pointer to location where data resides.

Instruction set of 8086:

Data transfer instructions

GENERAL – PURPOSE BYTE OR WORD TRANSFER INSTRUCTIONS:

MOV

PUSH

POP

XCHG

XLAT

SIMPLE INPUT AND OUTPUT PORT TRANSFER INSTRUCT

IN

OUT

SPECIAL ADDRESS TRANSFER INSTRUCTIONS

LEA

LDS

LES

FLAG TRANSFER INSTRUCTIONS:

LAHF

SAHF

PUSHF

POPF

ADDITION INSTRUCTIONS:

ADD

ADC

INC

AAA

DAA

SUBTRACTION INSTRUCTIONS:

SUB

SBB

DEC

NEG

CMP

AAS

DAS

MULTIPLICATION INSTRUCTIONS:

MUL

IMUL

AAM

DIVISION INSTRUCTIONS:

DIV

IDIV

AAD

CBW

CWD

BIT MANIPULATION INSTRUCTIONS

LOGICAL INSTRUCTIONS:

NOT

AND

OR

XOR

TEST

SHIFT INSTRUCTIONS:

SHL / SAL

SHR

SAR

PROGRAM EXECUTION TRANSFER INSTRUCTIONS

UNCONDITIONAL TRANSFER INSTRUCTIONS:

CALL

RET

JMP

CONDITIONAL TRANSFER INSTRUCTIONS:

JA / JNBE

JAE / JNB

JB / JNAE

JBE / JNA

JC
JE / JZ
JG / JNLE
JGE / JNL
JL / JNGE
JLE / JNG
JNC
JNE / JNZ
JNO
JNP / JPO
JNS
JO
JP / JPE
JS

ITERATION CONTROL INSTRUCTIONS:

LOOP
LOOPE / LOOPZ
LOOPNE / LOOPNZ
JCXZ

INTERRUPT INSTRUCTIONS:

INT
INTO
IRET

PROCESS CONTROL INSTRUCTIONS

FLAG SET / CLEAR INSTRUCTIONS:

STC
CLC
CMC
STD
CLD
STI

CLI

EXTERNAL HARDWARE SYNCHRONIZATION INSTRUCTIONS:

HLT

WAIT

ESC

LOCK

NOP

Instruction Description

AAA Instruction - ASCII Adjust after Addition

AAD Instruction - ASCII adjust before Division

AAM Instruction - ASCII adjust after Multiplication

AAS Instruction - ASCII Adjust for Subtraction

ADC Instruction - Add with carry.

ADD Instruction - ADD destination, source

AND Instruction - AND corresponding bits of two operands

Unit-III

8255-PPI:

The 8255 is a general purpose programmable I/O device used for parallel data transfer. It has 24 I/O pins which can be grouped in three 8-bit parallel ports : Port A, Port B and Port C. The eight bits of port C can be used as individual bits or be grouped in two 4-bit ports : C_{upper} (C_U) and C_{lower} (C_L).

The 8255, primarily, can be programmed in two basic modes : Bit Set/Reset (BSR) mode and I/O mode. The BSR mode is used to set or reset the bits in port C. The I/O mode is further divided into three modes :

- Mode 0 : Simple Input/Output
- Mode 1 : Input/Output with handshake
- Mode 2 : Bi-directional I/O data transfer

The function of I/O pins (input or output) and modes of operation of I/O ports can be programmed by writing proper control word in the control word register. Each bit in the control word has a specific meaning and the status of these bits decides the function and operating mode of the I/O ports.

Features of 8255A

1. The 8255A is a widely used, programmable, parallel I/O device.
2. It can be programmed to transfer data under various conditions, from simple I/O to interrupt I/O.
3. It is compatible with all Intel and most other microprocessors.
4. It is completely TTL compatible.
5. It has three 8-bit ports : Port A, Port B, and Port C, which are arranged in two groups of 12 pins. Each port has an unique address, and data can be read from or written to a port. In addition to the address assigned to the three ports, another address is assigned to the control register into which control words are written for programming the 8255 to operate in various modes.
6. Its bit set/reset mode allows setting and resetting of individual bits of Port C.
7. The 8255 can operate in 3 I/O modes : (i) Mode 0, (ii) Mode 1, and (iii) Mode 2.
 - a) In Mode 0, Port A and Port B can be configured as simple 8-bit input or output ports without handshaking. The two halves of Port C can be programmed separately as 4-bit input or output ports.
 - b) In Mode 1, two groups each of 12 pins are formed. Group A consists of Port A and the upper half of Port C while Group B consists of Port B and the lower half of Port C. Ports A and B can be programmed as 8-bit Input or Output ports with three lines of Port C in each group used for handshaking.
 - c) In Mode 2, only Port A can be used as a bidirectional port. The handshaking signals are provided on five lines of Port C ($PC_3 - PC_7$). Port B can be used in Mode 0 or in Mode 1.
8. All I/O pins of 8255 has 2.5 mA DC driving capacity (i.e. sourcing current of 2.5 mA).

Block Diagram

Fig. 7.2 shows the internal block diagram of 8255A. It consists of data bus buffer, control logic and Group A and Group B controls.

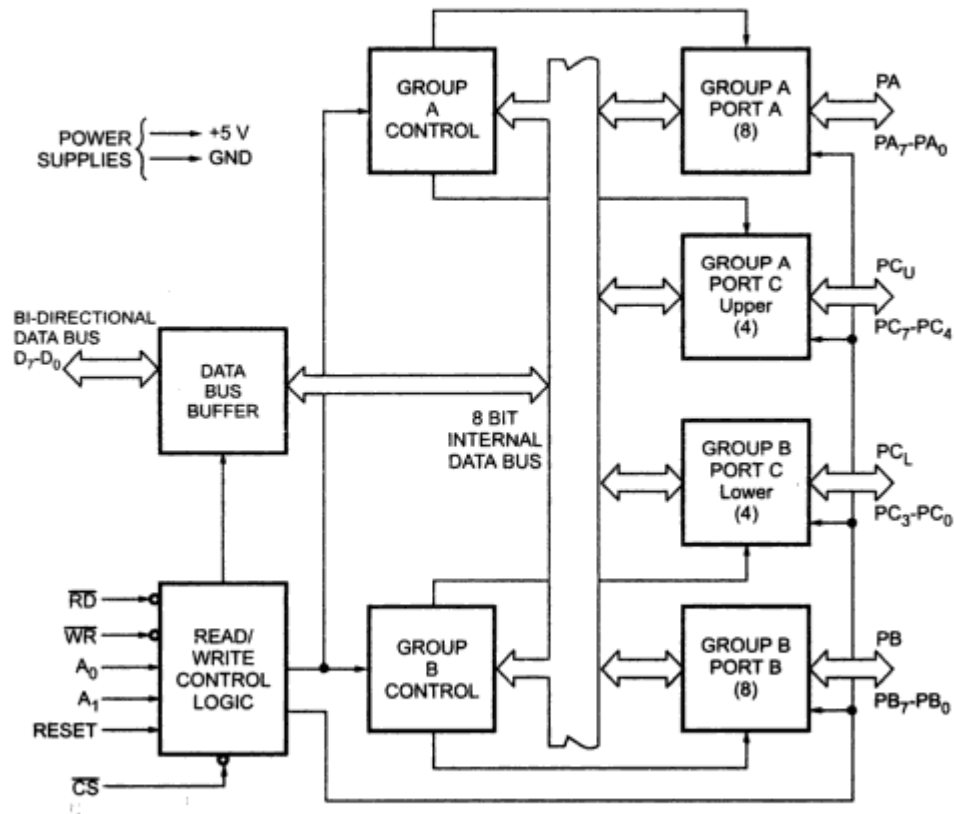


Fig. 7.2 Block diagram of 8255A

Mode 1 : Input/Output with handshake

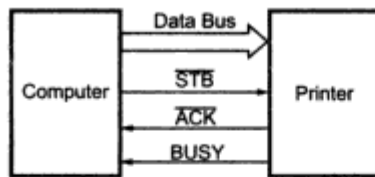


Fig. 7.3 Data transfer between computer and printer using handshaking signals

alongwith data signals. Fig. 7.3 shows data transfer between computer and printer using handshaking signals.

These handshaking signals are used to tell computer whether printer is ready to accept the data or not. If printer is ready to accept the data then after sending data on data bus, computer uses another handshaking signal (\overline{STB}) to tell printer that valid data is available on the data bus.

The 8255 mode 1 which supports handshaking has following features.

1. Two ports (A and B) function as 8-bit I/O ports. They can be configured either as input or output ports.
2. Each port uses three lines from Port C as handshake signals. The remaining two lines of Port C can be used for simple I/O functions.
3. Input and output data are latched.
4. Interrupt logic is supported.

Mode 2 : Bi-directional I/O data transfer

This mode allows bi-directional data transfer (transmission and reception) over a single 8-bit data bus using handshaking signals. This feature is available only in Group A with Port A as the 8-bit bi-directional data bus; and $PC_3 - PC_7$ are used for handshaking purpose. In this mode, both inputs and outputs are latched. Due to use of a single 8-bit data bus for bi-directional data transfer, the data sent out by the CPU through Port A appears on the bus connecting it to the peripheral, only when the peripheral requests it. The remaining lines of Port C i.e. $PC_0 - PC_2$ can be used for simple I/O functions. The Port B can be programmed in mode 0 or in mode 1. When Port B is programmed in mode 1, $PC_0 - PC_2$ lines of Port C are used as handshaking signals.

In this mode, input or output data transfer is controlled by handshaking signals. Handshaking signals are used to transfer data between devices whose data transfer speeds are not same. For example, computer can send data to the printer with large speed but printer can't accept data and print data with this rate. So computer has to send data with the speed with which printer can accept. This type of data transfer is achieved by using handshaking signals

7.5 Control Word Formats

A high on the RESET pin causes all 24 lines of the three 8-bit ports to be in the input mode. All flip-flops are cleared and the interrupts are reset. This condition is maintained even after the RESET goes low. The ports of the 8255 can then be programmed for any other mode by writing a single control word into the control register, when required.

For Bit Set/Reset Mode

Fig. 7.4 shows bit set/reset control word format.

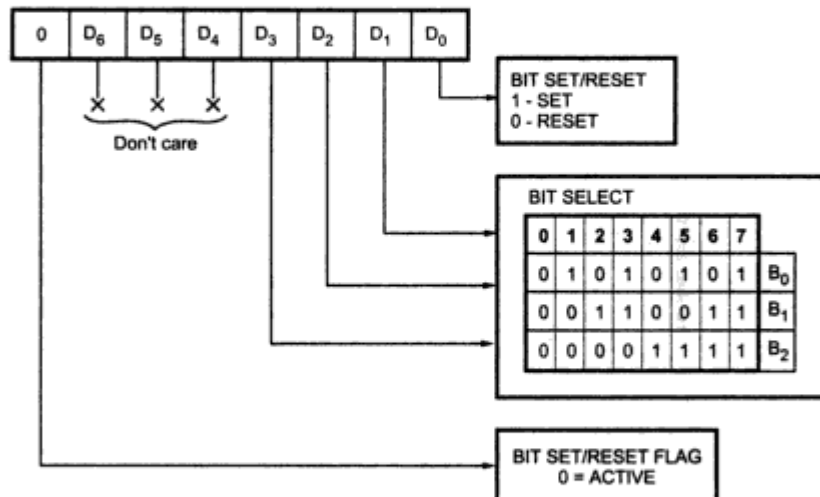


Fig. 7.4 Bit set/reset control word format

The eight possible combinations of the states of bits D₃ - D₁ (B₂ B₁ B₀) in the Bit Set-Reset format (BSR) determine particular bit in PC₀ - PC₇ being set or reset as per the status of bit D₀. A BSR word is to be written for each bit that is to be set or reset. For example, if bit PC₃ is to be set and bit PC₄ is to be reset, the appropriate BSR words that will have to be loaded into the control register will be, 0XXX0111 and 0XXX1000, respectively, where X is don't care.

The BSR word can also be used for enabling or disabling interrupt signals generated by Port C when the 8255 is programmed for Mode 1 or 2 operation. This is done by setting or resetting the associated bits of the interrupts. This is described in detail in next section.

For I/O Mode

The mode definition format for I/O mode is shown in Fig. 7.5. The control words for both, mode definition and Bit Set-Reset are loaded into the same control register, with bit D₇ used for specifying whether the word loaded into the control register is a mode

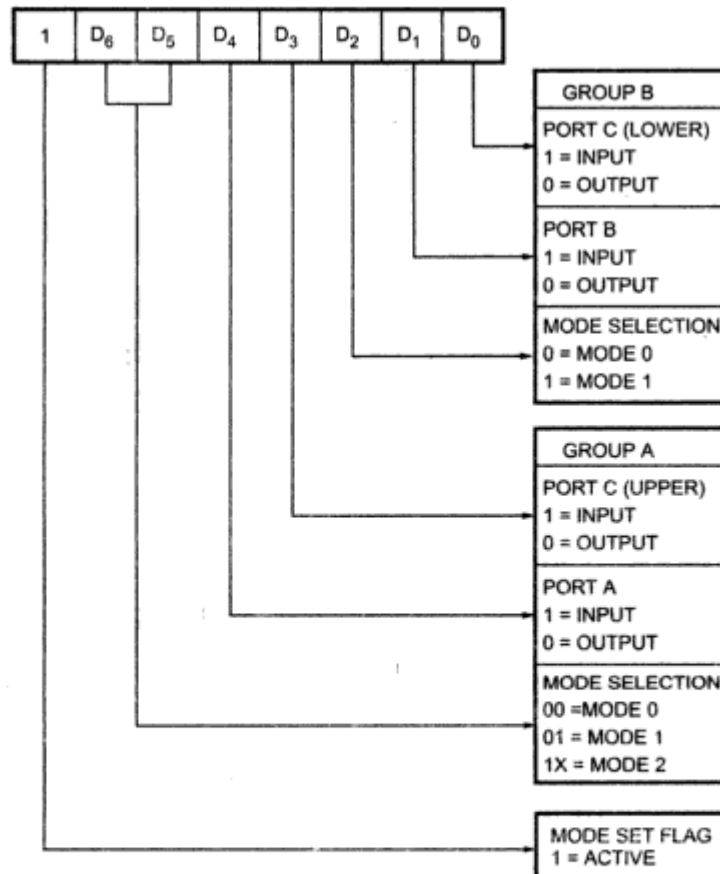


Fig. 7.5 8255 Mode definition format

definition word or Bit Set-Reset word. If D_7 is high, the word is taken as a mode definition word, and if it is low, it is taken as a Bit Set-Reset word. The appropriate bits are set or reset depending on the type of operation desired, and loaded into the control register.

Interfacing 8255 to 8086 in I/O Mapped I/O Mode

The 8086 has four special instructions IN, INS, OUT, and OUTS to transfer data through the input/output ports in I/O mapped I/O system. M/\overline{IO} signal is always low when 8086 is executing these instructions. So M/\overline{IO} signal is used to generate separate addresses for, memory and input/output. Only 256 (2^8) I/O addresses can be generated when direct addressing method is used. By using indirect address method this range can be extended upto 65536 (2^{16}) addresses.

Fig. 7.17 shows the interfacing of 8255 with 8086 in I/O mapped I/O technique. Here, \overline{RD} and \overline{WR} signals are activated when M/\overline{IO} signal is low, indicating I/O bus cycle. Only lower data bus ($D_0 - D_7$) is used as 8255 is 8-bit device. Reset out signal from clock generator is connected to the Reset signal of the 8255. In case of interrupt driven I/O INTR signal (PC_3 or PC_0) from 8255 is connected to INTR input of 8088.

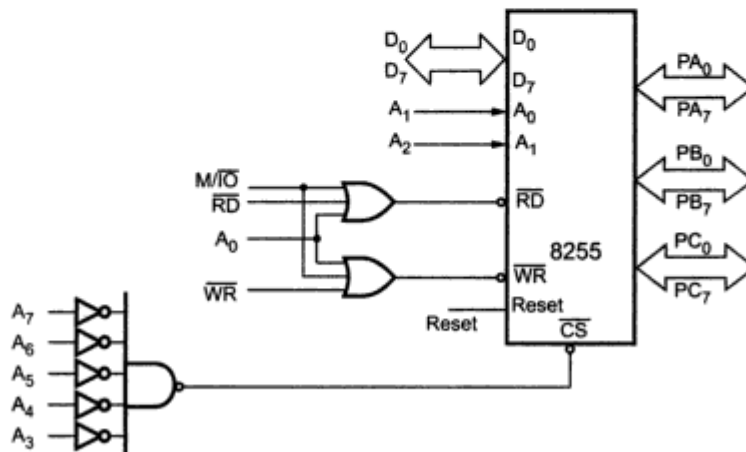


Fig. 7.17 I/O mapped I/O

I/O Map :

Port / control Register	Address lines								Address
	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
Port A	0	0	0	0	0	0	0	0	00H
Port B	0	0	0	0	0	0	1	0	02H
Port C	0	0	0	0	0	1	0	0	04H
Control register	0	0	0	0	0	1	1	0	06H

Note : It is assumed that the direct addressing is used.

Interfacing 8255 to 8086 in Memory Mapped I/O

In this type of I/O interfacing, the 8086 uses 20 address lines to identify an I/O device; an I/O device is connected as if it is a memory register. The 8086 uses same control signals and instructions to access I/O as those of memory. Fig. 7.18 shows the interfacing of 8255 with 8086 in memory mapped I/O technique. Here \overline{RD} and \overline{WR} signals are activated when M/\overline{IO} signal is high, indicating memory bus cycle. Address lines $A_0 - A_1$ are used by 8255 for internal decoding. To get absolute address, all remaining address lines ($A_3 - A_{19}$) are used to decode the address for 8255. Other signal connections are same as in I/O mapped I/O.

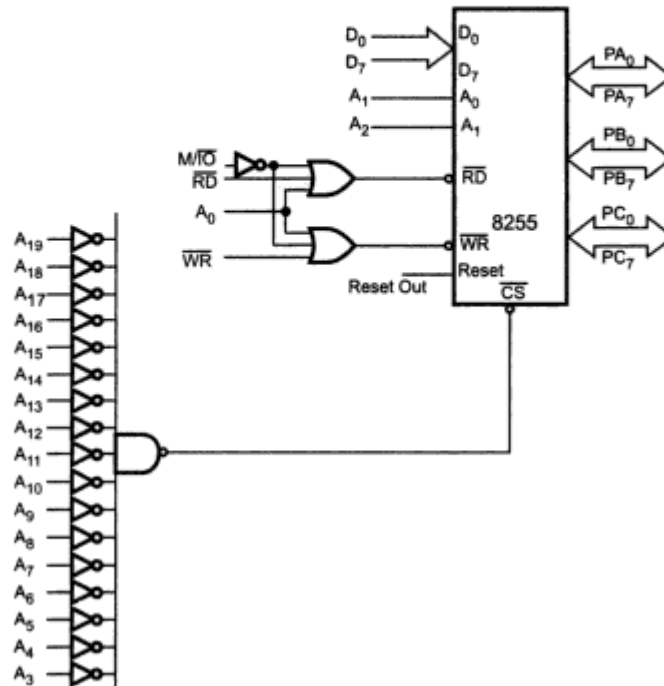


Fig. 7.18 Memory mapped I/O

I/O Map :

Register	A ₁₉	A ₁₈	A ₁₇	A ₁₆	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	Address
Port A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000H
Port B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	00002H
Port C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	00004H
Control register	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	00006H

ADC 0808/0809

The ADC 0808 and ADC 0809 are monolithic CMOS devices with an 8-channel multiplexer. These devices are also designed to operate from common microprocessor control buses, with tri-state output latches driving the data bus. The main features of these devices are :

Features

- 8-bit successive approximation ADC.
- 8-channel multiplexer with address logic.
- Conversion time 100 μ s.
- It eliminates the need for external zero and full-scale adjustments.
- Easy to interface to all microprocessors.
- It operates on single 5 V power supply.
- Output meet TTL voltage level specifications.

Pin Diagram

Fig. 7.37 shows pin diagram of 0808/0809 ADC.

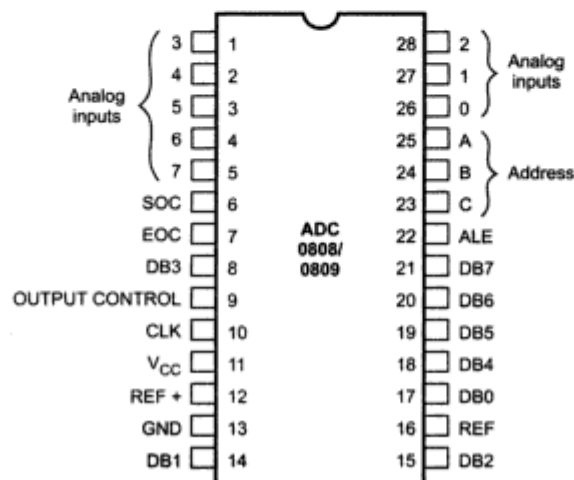


Fig. 7.37 Pin diagram of 0808/0809

Operation

ADC 0808/0809 has eight input channels, so to select desired input channel, it is necessary to send 3-bit address on A, B and C inputs. The address of the desired channel is sent to the multiplexer address inputs through port pins. After at least 50 ns, this address must be latched. This can be achieved by sending ALE signal. After another 2.5 μ s, the start of conversion (SOC) signal must be sent high and then low to start the conversion process. To indicate end of conversion ADC 0808/0809 activates EOC signal. The microprocessor system can read converted digital word through data bus by enabling the output enable signal after EOC is activated. This is illustrated in Fig. 7.38.

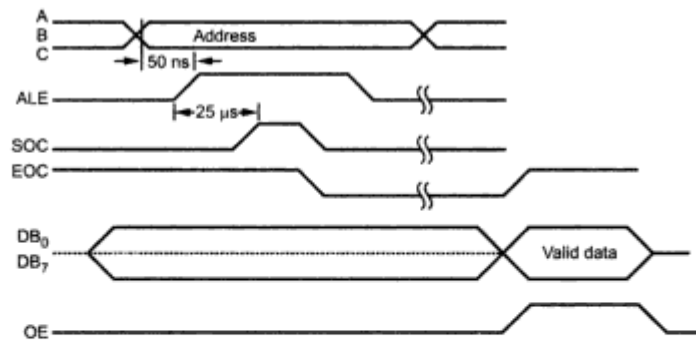


Fig. 7.38 Timing waveforms for ADC 0808

Interfacing

Fig. 7.39 shows typical interfacing circuit for ADC 0808 with microprocessor system.

The zener diode and LM 308 amplifier circuitry is used to produce a V_{CC} and $+V_{REF}$ of 5.12 V for the A/D converter. With this reference voltage the A/D converter will have 256 steps of 20 mV each.

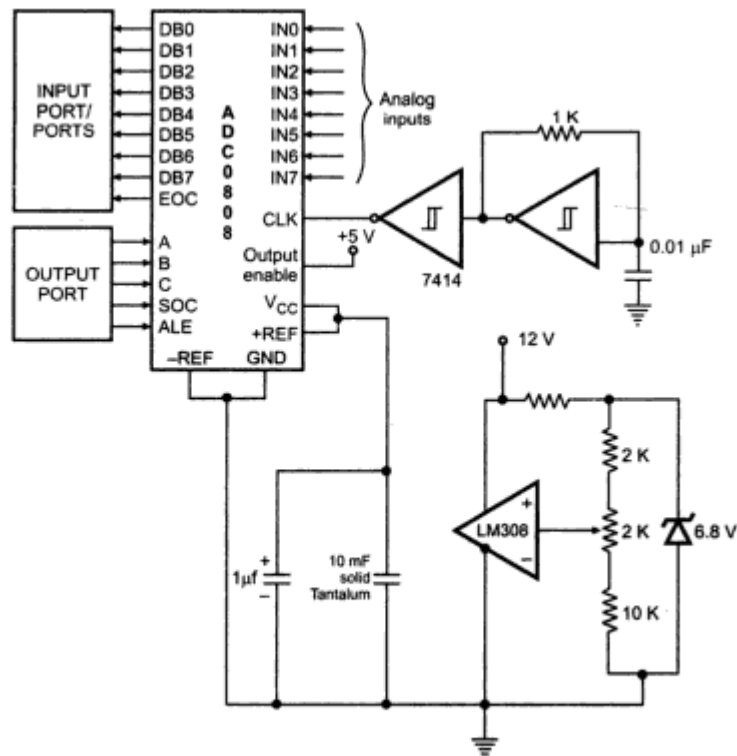


Fig. 7.39 Typical interface for 0808/0809

Stepper Motor Interfacing

A stepper motor is a digital motor. It can be driven by digital signal. Fig. 7.40 shows the typical 2 phase motor interfaced using 8255. Motor shown in the circuit has two phases, with center-tap winding. The center taps of these windings are connected to the 12 V supply. Due to this, motor can be excited by grounding four terminals of the two windings. Motor can be rotated in steps by giving proper excitation sequence to these windings. The lower nibble of port A of the 8255 is used to generate excitation signals in the proper sequence.

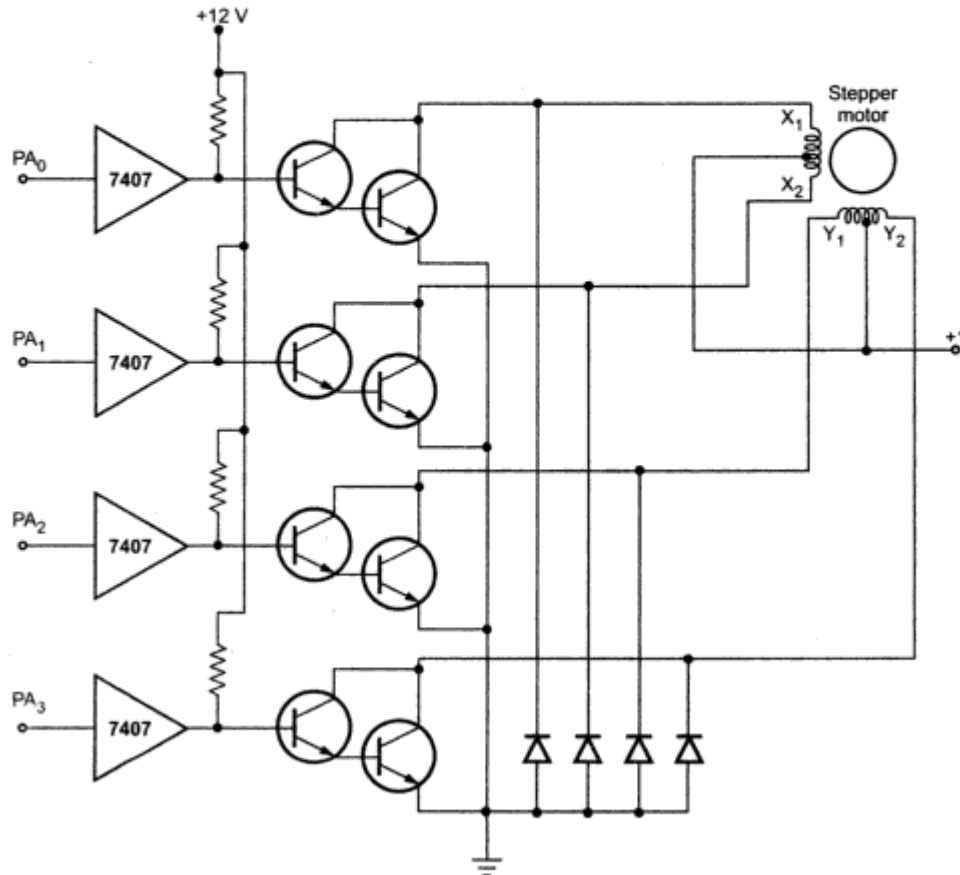


Fig. 7.40 Stepper motor interface

The Table 7.3 shows typical excitation sequence. The given excitation sequence rotates the motor in clockwise direction. To rotate motor in anticlockwise direction we have to excite motor in a reverse sequence. The excitation sequence for stepper motor may change due to change in winding connections. However, it is not desirable to excite both the ends

of the same winding simultaneously. This cancels the flux and motor winding may damage. To avoid this, digital locking system must be designed. Fig. 7.41 shows a simple digital locking system. Only one output is activated (made low) when properly excited; otherwise output is disabled (made high).

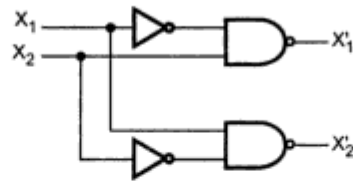


Fig. 7.41 Digital locking system

Step	X_1	X_2	Y_1	Y_2
1	0	1	0	1
2	1	0	0	1
3	1	0	1	0
4	0	1	1	0
1	0	1	0	1

Table 7.3 Full step excitation sequence

The excitation sequence given in Table 7.3 is called **full step sequence** in which excitation ends of the phase are changed in one step. The excitation sequence given in Table 7.4 takes two steps to change the excitation ends of the phase. Such a sequence is called **half step sequence** and in each step the motor is rotated by 0.9° .

Step	X_1	X_2	Y_1	Y_2
1	0	1	0	1
2	0	0	0	1
3	1	0	0	1
4	1	0	0	0
5	1	0	1	0
6	0	0	1	0
7	0	1	1	0
8	0	1	0	0
1	0	1	0	1

Table 7.4 Half step excitation sequence

We know that stepper motor is stepped from one position to the next by changing the currents through the fields in the motor. The winding inductance opposes the change in current and this puts limit on the stepping rate. For higher stepping rates and more torque, it is necessary to use a higher voltage source and current limiting resistors as shown in Fig. 7.42. By adding series resistance, we decrease L/R time constant, which allows the current to change more rapidly in the windings. There is a power loss across series resistor, but designer has to compromise between power and speed.

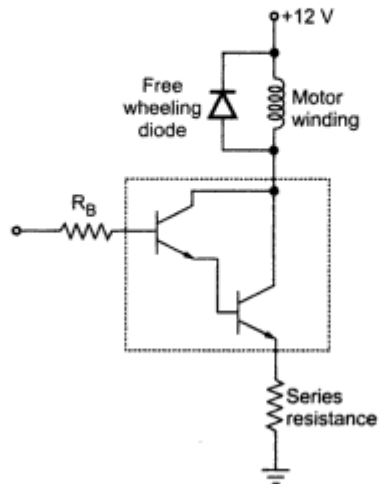


Fig. 7.42 Excitation circuit with series resistance

MEMORY INTERFACING:

- **Control connections:** A ROM usually has only one control input, while a RAM often has one or two control inputs.
- The control input most often found on the ROM is the output enable (OE) or gate (G), this allows data to flow out of the output data pins of the ROM.
- If OE and the selected input are both active, then the output is enable, if OE is inactive, the output is disabled at its high-impedance state.
- The OE connection enables and disables a set of three-state buffer located within the memory device and must be active to read data.
- A RAM memory device has either one or two control inputs. If there is one control input it is often called R/ W .
- This pin selects a read operation or a write operation only if the device is selected by the selection input (CS). If the RAM has two control inputs, they are usually labeled WE or W and OE or G .
- (WE) write enable must be active to perform a memory write operation and OE must be active to perform a memory read operation.
- When these two controls WE and OE are present, they must never be active at the same time.
- The ROM read only memory permanently stores programs and data and data was always present, even when power is disconnected.
- It is also called as nonvolatile memory.
- EPROM (erasable programmable read only memory) is also erasable if exposed to high intensity ultraviolet light for about 20 minutes or less, depending upon the type of EPROM.
- We have PROM (programmable read only memory)
- RMM (read mostly memory) is also called the flash memory.
- The flash memory is also called as an EEPROM (electrically erasable programmable ROM), EAROM (electrically alterable ROM), or a NOVRAM (nonvolatile ROM).
- These memory devices are electrically erasable in the system, but require more time to erase than a normal RAM.
- EPROM contains the series of 27XXX contains the following part numbers :

2704(512 * 8), 2708(1K * 8), 2716(2K * 8), 2732(4K * 8), 2764(8K * 8), 27128(16K * 8) etc..

- Each of these parts contains address pins, eight data connections, one or more chip selection inputs (CE) and an output enable pin (OE).
- This device contains **11** address inputs and **8** data outputs.
- If both the pin connection CE and OE are at logic 0, data will appear on the output connection . If both the pins are not at logic 0, the data output connections remains at their high impedance or off state.
- To read data from the EPROM Vpp pin must be placed at a logic 1.

Static RAM Interfacing

The semiconductor RAM is broadly two types – Static RAM and Dynamic RAM.

- The semiconductor memories are organized as two dimensional arrays of memory locations.
- For example 4K * 8 or 4K byte memory contains 4096 locations, where each locations contains 8-bit data and only one of the 4096 locations can be selected at a time. Once a location is selected all the bits in it are accessible using a group of conductors called Data bus.
- For addressing the 4K bytes of memory, 12 address lines are required.
- In general to address a memory location out of N memory locations, we will require at least n bits of address, i.e. n address lines where $n = \log_2 N$.
- Thus if the microprocessor has n address lines, then it is able to address at the most N locations of memory, where $2^n = N$. If out of N locations only P memory locations are to be interfaced, then the least significant p address lines out of the available n lines can be directly connected from the microprocessor to the memory chip while the remaining (n-p) higher order address lines may be used for address decoding as inputs to the chip selection logic.
- The memory address depends upon the hardware circuit used for decoding the chip select (CS). The output of the decoding circuit is connected with the CS pin of the memory chip.
- The general procedure of static memory interfacing with 8086 is briefly described as follows:
 1. Arrange the available memory chip so as to obtain 16- bit data bus width. The upper 8-bit bank is called as odd address memory bank and the lower 8-bit bank is called as

even address memory bank.

2. Connect available memory address lines of memory chip with those of the microprocessor and also connect the memory RD and WR inputs to the corresponding processor control signals. Connect the 16-bit data bus of the memory bank with that of the microprocessor 8086.

3. The remaining address lines of the microprocessor, BHE and A0 are used for decoding the required chip select signals for the odd and even memory banks. The CS of memory is derived from the o/p of the decoding circuit.

- As a good and efficient interfacing practice, the address map of the system should be continuous as far as possible, i.e. there should not be no windows in the map and no fold back space should be allowed. A memory location should have a single address corresponding to it, i.e. absolute decoding should be preferred and minimum hardware should be used for decoding large capacity memory is required in a microcomputer system, the memory subsystem is generally designed using dynamic RAM because there are various advantages of dynamic RAM.

- E.g. higher packing density, lower cost and less power consumption. A typical static RAM cell may require six transistors while the dynamic RAM cell requires only a transistors along with a capacitor. Hence it is possible to obtain higher packaging density and hence low cost units are available.

- The basic dynamic RAM cell uses a capacitor to store the charge as a representation of data. This capacitor is manufactured as a diode that is reverse-biased so that the storage capacitance comes into the picture.

- This storage capacitance is utilized for storing the charge representation of data but the reverse-biased diode has leakage current that tends to discharge the capacitor giving rise to the possibility of data loss. To avoid this possible data loss, the data stored in a dynamic RAM cell must be refreshed after a fixed time interval regularly. The process of refreshing the data in RAM is called as

Refresh cycle.

- The refresh activity is similar to reading the data from each and every cell of memory, independent of the requirement of microprocessor. During this refresh period all other operations related to the memory subsystem are suspended. Hence the refresh activity causes loss of time, resulting in reduce system performance.

- However keeping in view the advantages of dynamic RAM, like low power consumption, high packaging density and low cost, most of the advanced computing system are designed using dynamic RAM, at the cost of operating speed.

- A dedicated hardware chip called as dynamic RAM controller is the most important part of the interfacing circuit.

- The **Refresh cycle** is different from the memory read cycle in the following aspects.

1. The memory address is not provided by the CPU address bus, rather it is generated by a refresh mechanism counter called as refresh counter.

2. Unlike memory read cycle, more than one memory chip may be enabled at a time so as to reduce the number of total memory refresh cycles.

3. The data enable control of the selected memory chip is deactivated, and data is not allowed to appear on the system data bus during refresh, as more than one memory units are refreshed simultaneously. This is to avoid the data from the different chips to appear on the bus simultaneously.

4. Memory read is either a processor initiated or an external bus master initiated and carried out by the refresh mechanism. Dynamic RAM is available in units of several kilobits to megabits of memory.

This memory is arranged internally in a two dimensional matrix array so that it will have n rows and m columns. The row address n and column address m are important for the refreshing operation.

- For example, a typical 4K bit dynamic RAM chip has an internally arranged bit array of dimension $64 * 64$, i.e. 64 rows and 64 columns. The row address and column address will require 6 bits each. These 6 bits for each row address and column address will be generated by the refresh counter, during the refresh cycles.

- A complete row of 64 cells is refreshed at a time to minimize the refreshing time.

Thus the refresh counter needs to generate only row addresses. The row address are multiplexed, over lower order address lines.

- The refresh signals act to control the multiplexer, i.e. when refresh cycle is in process the refresh counter puts the row address over the address bus for refreshing.

Otherwise, the address bus of the processor is connected to the address bus of DRAM, during normal processor initiated activities.

- A timer, called refresh timer, derives a pulse for refreshing action after each refresh

interval.

- Refresh interval can be qualitatively defined as the time for which a dynamic RAM cell can hold data charge level practically constant, i.e. no data loss takes place.

- Suppose the typical dynamic RAM chip has 64 rows, then each row should be refreshed after each refresh interval or in other words, all the 64 rows are to be refreshed in a single refresh interval.

- This refresh interval depends upon the manufacturing technology of the dynamic RAM cell. It may range anywhere from 1ms to 3ms.

- Let us consider 2ms as a typical refresh time interval. Hence, the frequency of the refresh pulses will be calculated as follows:

- Refresh Time (per row) $t_r = (2 \times 10^{-3}) / 64$.

- Refresh Frequency $f_r = 64 / (2 \times 10^{-3}) = 32 \times 10^3 \text{ Hz}$.

- The following block diagram explains the refreshing logic and 8086 interfacing with dynamic RAM.

- Each chip is of 16K * 1-bit dynamic RAM cell array. The system contains two 16K byte dynamic RAM units. All the address and data lines are assumed to be available from an 8086 microprocessor system.

- The OE pin controls output data buffer of the memory chips. The CE pins are active high chip selects of memory chips. The refresh cycle starts, if the refresh output of the refresh timer goes high, OE and CE also tend to go high.

- The high CE enables the memory chip for refreshing, while high OE prevents the data from appearing on the data bus, as discussed in memory refresh cycle. The 16K * 1-bit dynamic RAM has an internal array of 128*128 cells, requiring 7 bits for row address. The lower order seven lines A₀-A₆ are multiplexed with the refresh counter output A₁₀-A₁₆.

- If the RAM has two control inputs, they are usually labeled WE or W and OE or G.

- (WE) write enable must be active to perform a memory write operation and OE must be active to perform a memory read operation.

- When these two controls WE and OE are present, they must never be active at the same time

- The ROM read only memory permanently stores programs and data and data was always present, even when power is disconnected.

- It is also called as nonvolatile memory.

- EPROM (erasable programmable read only memory) is also erasable if exposed to high intensity ultraviolet light for about 20 minutes or less, depending upon the type of EPROM.

- We have PROM (programmable read only memory)

- RMM (read mostly memory) is also called the flash memory.

- The flash memory is also called as an EEPROM (electrically erasable programmable ROM), EAROM (electrically alterable ROM), or a NOVRAM (nonvolatile ROM).

- These memory devices are electrically erasable in the system, but require more time to erase than a normal RAM.

- EPROM contains the series of 27XXX contains the following part numbers :

2704(512 * 8), 2708(1K * 8), 2716(2K * 8), 2732(4K * 8), 2764(8K * 8), 27128(16K * 8) etc..

- Each of these parts contains address pins, eight data connections, one or more chip selection inputs (CE) and an output enable pin (OE).

- This device contains **11** address inputs and **8** data outputs.

- If both the pin connection CE and OE are at logic 0, data will appear on the output connection . If both the pins are not at logic 0, the data output connections remains at their high impedance or off state.

- To read data from the EPROM Vpp pin must be placed at a logic 1.

Interrupt Cycle of 8086/88

An 8086 interrupt can come from any one the three sources :

- External signal
- Special Instruction in the program
- Condition produced by instruction

8.2.1 External Signal (Hardware Interrupt)

An 8086 can get interrupt from an external signal applied to the nonmaskable interrupt (NMI) input pin, or the interrupt (INTR) input pin.

8.2.2 Special Instruction

8086 supports a special instruction, INT to execute special program. At the end of the interrupt service routine, execution is usually returned to the interrupted program.

8.2.3 Condition Produced by Instruction

An 8086 is interrupted by some condition produced in the 8086 by the execution of an instruction. For example divide by zero : Program execution will automatically be interrupted if you attempt to divide an operand by zero.

At the end of each instruction cycle 8086 checks to see if there is any interrupt request. If so, 8086 responds to the interrupt by performing series of actions (Refer Fig. 8.1).

1. It decrements stack pointer by 2 and pushes the flag register on the stack .
2. It disables the INTR interrupt input by clearing the interrupt flag in the flag register.
3. It resets the trap flag in the flag register.
4. It decrements stack pointer by 2 and pushes the current code segment register contents on the stack.
5. It decrements stack pointer by 2 and pushes the current instruction pointer contents on the stack.
6. It does an indirect far jump at the start of the procedure by loading the CS and IP values for the start of the interrupt service routine (ISR).

An IRET instruction at the end of the interrupt service procedure returns execution to the main program.

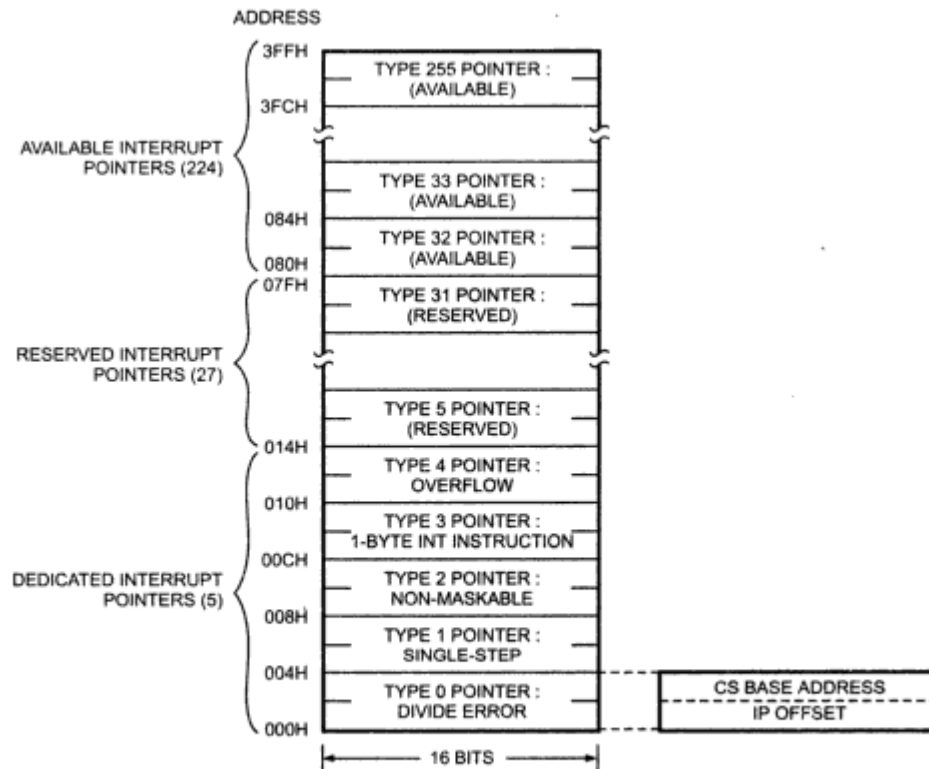


Fig. 8.2 8086 interrupt vector table

8086 Interrupt Types

Divide by Zero Interrupt (Type 0)

When the quotient from either a DIV or IDIV instruction is too large to fit in the result gister; 8086 will automatically execute type 0 interrupt.

Single Step Interrupt (Type 1)

The type 1 interrupt is the single step trap. In the single step mode, system will execute one instruction and wait for further direction from user. Then user can examine the contents of registers and memory locations and if they are correct, user can tell the system to execute the next instruction. This feature is useful for debugging assembly language programs.

Software Interrupts

Type 0 - 255 :

The 8086 INT instruction can be used to cause the 8086 to do one of the 256 possible interrupt types. The interrupt type is specified by the number as a part of the instruction. You can use an INT2 instruction to send execution to an NMI interrupt service routine. This allows you to test the NMI routine without needing to apply an external signal to the NMI input of the 8086.

With the software interrupts you can call the desired routines from many different programs in a system e.g. BIOS in IBM PC. The IBM PC has in its ROM collection of routines, each performing some specific function such as reading character from keyboard, writing character to CRT. This collection of routines referred to as **Basic Input Output System** or **BIOS**.

The BIOS routines are called with INT instructions. We will summarize interrupt response and how it is serviced by going through following steps.

1. 8086 pushes the flag register on the stack.
2. It disables the single step and the INTR input by clearing the trap flag and interrupt flag in the flag register.
3. It saves the current CS and IP register contents by pushing them on the stack.
4. It does an indirect far jump to the start of the routine by loading the new values of CS and IP register from the memory whose address calculated by multiplying 4 to the interrupt type. For example, if interrupt type is 4 then memory address is $4 \times 4 = 10_{10} = 10H$. So 8086 will read new value of IP from 00010H and CS from 00012H.
5. Once these values are loaded in the CS and IP, 8086 will fetch the instruction from the new address which is the starting address of interrupt service routine.
6. An IRET instruction at the end of the interrupt service routine gets the previous values of CS and IP by popping the CS and IP from the stack.
7. At the end the flag register contents are copied back into flag register by popping the flag register from stack.

Features of 8259

1. It can manage eight priority interrupts. This is equivalent to provide eight interrupt pins on the processor in place of INTR pin.
2. It is possible to locate vector table for these additional interrupts any where in the memory map. However, all eight interrupts are spaced at the interval of either four or eight locations.
3. By cascading 8259s it is possible to get 64 priority interrupts.
4. Interrupt mask register makes it possible to mask individual interrupt request.
5. The 8259A can be programmed to accept either the level triggered or the edge triggered interrupt request.
6. With the help of 8259A user can get the information of pending interrupts, in-service interrupts and masked interrupts.
7. The 8259A is designed to minimize the software and real time overhead in handling multi-level priority interrupts.

Block Diagram of 8259A

Fig. 8.6 shows the internal block diagram of the 8259A. It includes eight blocks : data bus buffer, read/write logic, control logic, three registers (IRR, ISR and IMR), priority resolver, and cascade buffer.

Data Bus Buffer

The data bus allows the 8086 to send control words to the 8259A and read a status word from the 8259A and read a status word from the 8259A. The 8-bit data bus also allows the 8259A to send interrupt types to the 8086.

Priority Resolver

The priority resolver determines the priorities of the bits set in the IRR. The bit corresponding to the highest priority interrupt input is set in the ISR during the $\overline{\text{INTA}}$ input.

Cascade Buffer Comparator

This section generates control signals necessary for cascade operations. It also generates Buffer-Enable signals. As stated earlier, the 8259 can be cascaded with other 8259s in order to expand the interrupt handling capacity to sixty-four levels. In such a case, the former is called a **master**, and the latter are called **slaves**. The 8259 can be set up as a master or a slave by the $\overline{\text{SP}} / \overline{\text{EN}}$ pin.

CAS 0 - 2

For a master 8259, the $\text{CAS}_0\text{-CAS}_2$ pins are outputs, and for slave 8259s, these are inputs. When the 8259 is a master (that is, when it accepts interrupt requests from other 8259s), the CALL opcode is generated by the Master in response to the first $\overline{\text{INTA}}$. The vectoring address must be released by the slave 8259. The master sends an identification code of three-bits (to select one out of the eight possible slave 8259s) on the $\text{CAS}_0\text{-CAS}_2$ lines. The slave 8259s accept these three signals as inputs (on their $\text{CAS}_0\text{-CAS}_2$ pins) and compare the code sent by the master with the codes assigned to them during initialisation. The slave thus selected (which had originally placed an interrupt request to the master 8259) then puts out the address of the interrupt service routine during the second and third $\overline{\text{INTA}}$ pulses from the CPU.

$\overline{\text{SP}} / \overline{\text{EN}}$ (Slave Program /Enable Buffer)

The $\overline{\text{SP}} / \overline{\text{EN}}$ signal is tied high for the master. However, it is grounded for the slave.

In large systems where buffers are used to drive the data bus, the data sent by the 8259 in response to $\overline{\text{INTA}}$ cannot be accessed by the CPU (due to the data bus buffer being disabled).

If an 8259 is used in the buffered mode (buffered or non-buffered modes of operation can be specified at the time of initialising the 8259), the $\overline{\text{SP}} / \overline{\text{EN}}$ pin is used as an output which can be used to enable the system data bus buffer whenever the 8259's data bus outputs are enabled (when it is ready to send data).

Means, in non-buffered mode, the $\overline{\text{SP}}/\overline{\text{EN}}$ pin of an 8259 is used to specify whether the 8259 is to operate as a master or as a slave, and in the buffered mode, the $\overline{\text{SP}}/\overline{\text{EN}}$ pin is used as an output to enable the data bus buffer of the system.

8.5.3 Interrupt Sequence

The events occur as follows in an 8086 system :

1. One or more of the INTERRUPT REQUEST lines (IR0-IR7) are raised high, setting the corresponding IRR bit(s).

2. The priority resolver checks three registers : The IRR for interrupt requests, the IMR for masking bits, and the ISR for the interrupt request being served. It resolves the priority and sets the INT high when appropriate.
3. The CPU acknowledges the INT and responds with an $\overline{\text{INTA}}$ pulse.
4. Upon receiving an $\overline{\text{INTA}}$ from the CPU, the highest priority ISR bit is set and the corresponding IRR bit is reset. The 8259A does not drive data bus during this cycle.
5. A selection of priority modes is available to the programmer so that the manner in which the requests are processed by the 8259A can be configured to match his system requirements. The priority modes can be changed or reconfigured dynamically at any time during the main program. This means that the complete interrupt service structure can be defined as required, based on the total system environment.
6. The 8086 will initiate a second INTA pulse. During this pulse, the 8259A releases a 8-bit pointer (interrupt type) onto the Data Bus where it is read by the CPU.
7. This completes the interrupt cycle. In the AEIOI mode the ISR bit is reset at the end of the second INTA pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

8.5.4 Priority Modes and Other Features

The various modes of operation of the 8259 are :

- (a) Fully Nested Mode,
- (b) Rotating Priority Mode,
- (c) Special Masked Mode, and
- (d) Polled Mode.

a) Fully Nested Mode :

After initialization, the 8259A operates in fully nested mode so it is called as default mode. The 8259 continues to operate in the Fully Nested Mode until the mode is changed through Operation Command Words. In this mode, IR0 has highest priority and IR7 has lowest priority. When the interrupt is acknowledged, it sets the corresponding bit in ISR. This bit will prevent all interrupts of the same or lower level, however it will accept higher priority interrupt requests. The vector address corresponding to this interrupt is then sent. The bit in the ISR will remain set until an EOI command is issued by the microprocessor at the end of interrupt service routine.

But if AEIOI (Automatic End of Interrupt) bit is set, the bit in the ISR resets at the trailing edge of the last $\overline{\text{INTA}}$.

(i) Automatic Rotation

In this mode, a device, after being serviced, receives the lowest priority. Assuming that IR₃ has just been serviced, it will receive the seventh priority.

IR ₀	IR ₁	IR ₂	IR ₃	IR ₄	IR ₅	IR ₆	IR ₇
4	5	6	7	0	1	2	3

(ii) Specific Rotation

In the Automatic Rotation mode, the interrupt request last serviced is assigned the lowest priority, whereas in the Specific Rotation mode, the lowest priority can be assigned to any interrupt input (IR₀ to IR₇) thus fixes all other priorities.

For example if the lowest priority is assigned to IR₂, other priorities are as shown below.

IR ₀	IR ₁	IR ₂	IR ₃	IR ₄	IR ₅	IR ₆	IR ₇
5	6	7	0	1	2	3	4

d) Special Mask Mode :

If any interrupt is in service then the corresponding bit is set in ISR and the lower priority interrupts are inhibited. Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution under software control, for example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion. In these cases we have to go for special mask mode.

In the special mask mode it inhibits further interrupts at that level and enables interrupts from all other levels (lower as well as higher) that are not masked. Thus any interrupt may be selectively enabled by loading the mask register.

e) Poll Mode :

In this mode the INT output is not used. The microprocessor checks the status of interrupt requests by issuing poll command. The microprocessor reads contents of 8259A after issuing poll command. During this read operation the 8259A provides polled word and sets ISR bit of highest priority active interrupt request FORMAT.

I	X	X	X	X	W ₂	W ₁	W ₀
---	---	---	---	---	----------------	----------------	----------------

I = 1 → One or more interrupt requests activated.

I = 0 → No interrupt request activated.

W₂ W₁ W₀ → Binary code of highest priority active interrupt request.

Programming the 8259A

The 8259A requires two types of command words. Initialization Command Words (ICWs) and Operational Command Words (OCWs).

The 8259A can be initialized with four ICWs; the first two are compulsory, and the other two are optional based on the modes being used. These words must be issued in a given sequence. After initialization, the 8259A can be set up to operate in various modes by using three different OCWs; however, they no longer need to be issued in a specific sequence.

Flow chart :

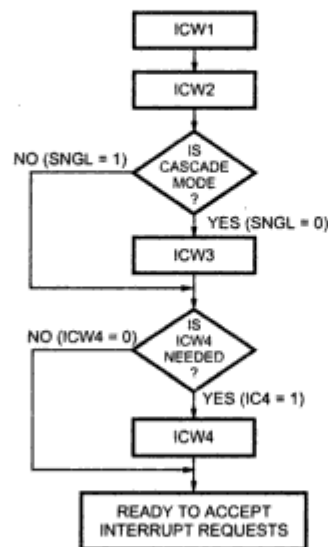


Fig. 8.7 8259 A initialization flowchart

Initialization Command Word 1 (ICW1)

Fig. 8.8 shows the Initialization Command Word 1 (ICW1).

A write command issued to the 8259 with $A_0 = 0$ and $D_4 = 1$ is interpreted as ICW1, which starts the initialization sequence.

It specifies

1. Single or multiple 8259As in the system.
2. 4 or 8 bit interval between the interrupt vector locations.
3. The address bits $A_7 - A_5$ of the CALL instruction.
4. Edge triggered or level triggered interrupts.
5. ICW4 is needed or not.

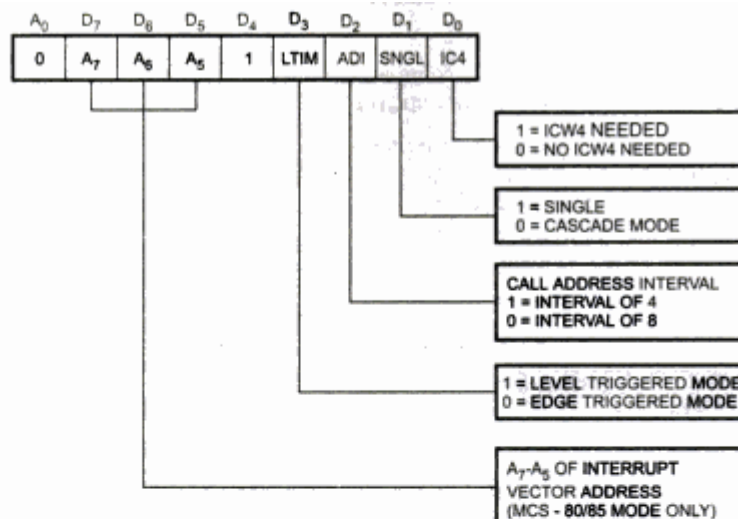


Fig. 8.8 Initialization command word 1 (ICW1)

Initialization Command Word 2 (ICW2)

Fig 8.9 shows the Initialization Command Word 2 (ICW2).

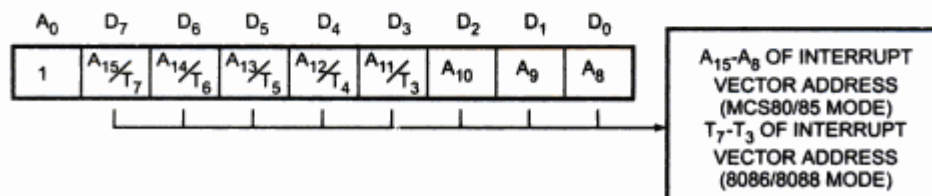


Fig. 8.9 Initialization command word 2 (ICW2)

A write command following ICW1, with A₀ = 1 is interpreted as ICW2. This is used to load the high order byte of the interrupt vector address of all the interrupts.

Initialization Command Word 3 (ICW3)

ICW3 is required only if there is more than one 8259 in the system and if they are cascaded. An ICW3 operation loads a slave register in the 8259. The format of the byte to be loaded as an ICW3 for a master 8259 or a slave is shown in the Fig. 8.10. For master, each bit in ICW3 is used to specify whether it has a slave 8259 attached to it on its corresponding IR (Interrupt Request) input. For slave, bits D₀-D₂ of ICW3 are used to assign a slave identification code (slave ID) to the 8259.

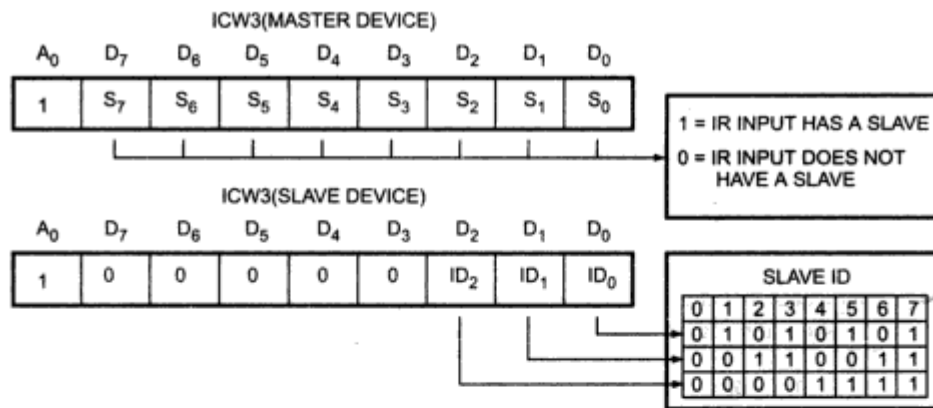


Fig. 8.10 Initialization command word 3 (ICW3)

Initialization Command Word 4 (ICW4)

It is loaded only if the D₀ bit of ICW1 (IC 4) is set. The format of ICW4 is shown in Fig. 8.11.

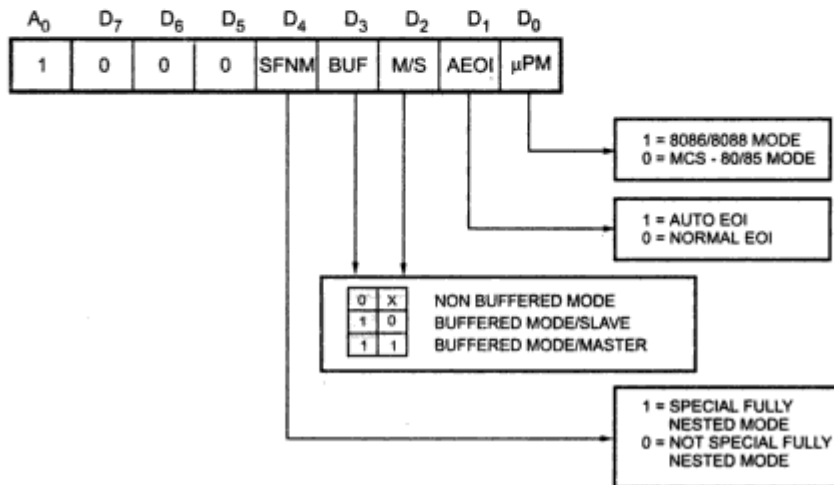


Fig. 8.11 Initialization command word 4 (ICW4)

It specifies.

- 1) Whether to use special fully nested mode or non special fully nested mode.
- 2) Whether to use buffered mode or non buffered mode.
- 3) Whether to use Automatic EOI or Normal EOI
- 4) CPU used, 8086/8088 or 80810.

SERIAL COMMUNICATION:

Classification

Serial data transmission can be classified on the basis of how transmission occurs.

1. Simplex
2. Half duplex
3. Full duplex

Simplex

In simplex, the hardware exists such that data transfer takes place only in one direction. There is no possibility of data transfer in the other direction. A typical example is transmission from a computer to the printer.

10.1.2 Half Duplex

The half duplex transmission allows the data transfer in both directions, but not simultaneously. A typical example is a walkie-talkie.

10.1.3 Full Duplex

The full duplex transmission allows the data transfer in both directions simultaneously. The typical example is transmission through telephone lines.

10.2 Transmission Formats

The data in the serial communication may be sent in two formats :

- a) Asynchronous
- b) Synchronous

10.2.1 Asynchronous

Fig. 10.1 shows the transmission format for asynchronous transmission. Asynchronous formats are character oriented. In this, the bits of a character or data word are sent at a constant rate, but characters can come at any rate (asynchronously) as long as they do not overlap. When no characters are being sent, a line stays high at logic 1 called **mark**, logic 0 is called **space**. The beginning of a character is indicated by a start bit which is always low. This is used to synchronize the transmitter and receiver. After the start bit, the data bits are sent with least significant bit first, followed by one or more stop bits (active high). The stop bits indicate the end of character. Different systems use 1, 1 1/2 or 2 stop bits. The combination of start bit, character and stop bits is known as **frame**. The start and stop bits carry no information, but are required because of the asynchronous nature of data. Fig. 10.2 illustrates how the data byte CAH would look when transmitted in the asynchronous serial format.

10.1.2 Half Duplex

The half duplex transmission allows the data transfer in both directions, but not simultaneously. A typical example is a walkie-talkie.

10.1.3 Full Duplex

The full duplex transmission allows the data transfer in both directions simultaneously. The typical example is transmission through telephone lines.

10.2 Transmission Formats

The data in the serial communication may be sent in two formats :

- a) Asynchronous b) Synchronous

10.2.1 Asynchronous

Fig. 10.1 shows the transmission format for asynchronous transmission. Asynchronous formats are character oriented. In this, the bits of a character or data word are sent at a constant rate, but characters can come at any rate (asynchronously) as long as they do not overlap. When no characters are being sent, a line stays high at logic 1 called **mark**, logic 0 is called **space**. The beginning of a character is indicated by a start bit which is always low. This is used to synchronize the transmitter and receiver. After the start bit, the data bits are sent with least significant bit first, followed by one or more stop bits (active high). The stop bits indicate the end of character. Different systems use 1, 1 1/2 or 2 stop bits. The combination of start bit, character and stop bits is known as **frame**. The start and stop bits carry no information, but are required because of the asynchronous nature of data. Fig. 10.2 illustrates how the data byte CAH would look when transmitted in the asynchronous serial format.

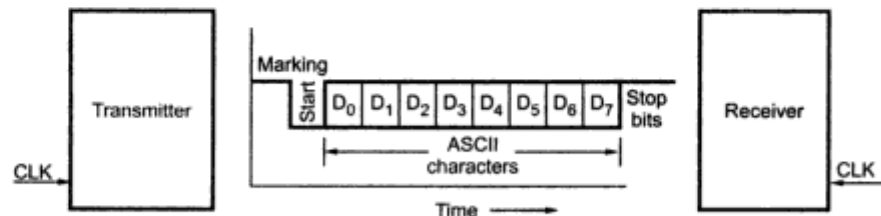


Fig. 10.1 Transmission format for asynchronous transmission

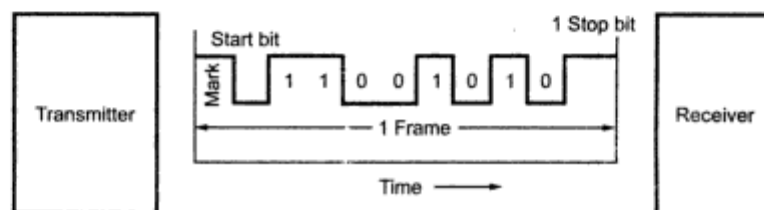


Fig. 10.2 Asynchronous format with data byte CAH

Copyrighted material

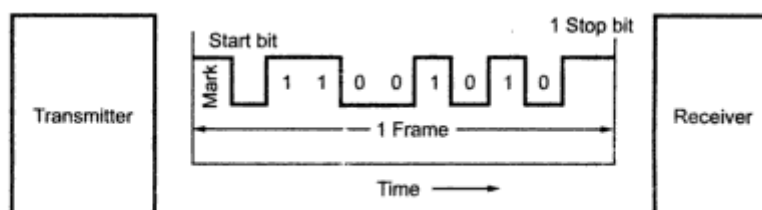


Fig. 10.2 Asynchronous format with data byte CAH

The data rate can be expressed as bits/sec. or characters/sec. The term bits/sec is also called the **baud rate**. The asynchronous format is generally used in low-speed transmission (less than 20 Kbits/sec).

10.2.2 Synchronous

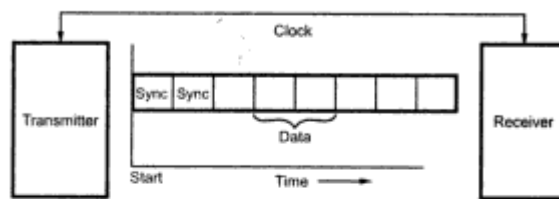


Fig. 10.3 Synchronous transmission format

The start and stop bits in each frame of asynchronous format represents wasted overhead bytes that reduce the overall character rate. These start and stop bits can be eliminated by synchronizing receiver and transmitter. They can be synchronized by having a common clock signal. Such a

communication is called **synchronous serial communication**. The Fig. 10.3 shows the transmission format of synchronous serial communication. In this transmission synchronous bits are inserted instead of start and stop bits.

Sr. No.	Asynchronous Serial Communication	Synchronous Serial Communication
1.	Transmitters and receivers are not synchronized by clock.	Transmitter and receivers are synchronized by clock.
2.	Bits of data are transmitted at constant rate.	Data bits are transmitted with synchronisation of clock.
3.	Character may arrive at any rate at receiver.	Character is received at constant rate.
4.	Data transfer is character oriented.	Data transfer takes place in blocks.
5.	Start and stop bits are required to establish communication of each character.	Start and stop bits are not required to establish communication of each character; however, synchronisation bits are required to transfer the data block.
6.	Used in low-speed transmissions at about speed less than 20 kbits/sec.	Used in high-speed transmissions

Table 10.1 Comparison between asynchronous and synchronous serial data transfer

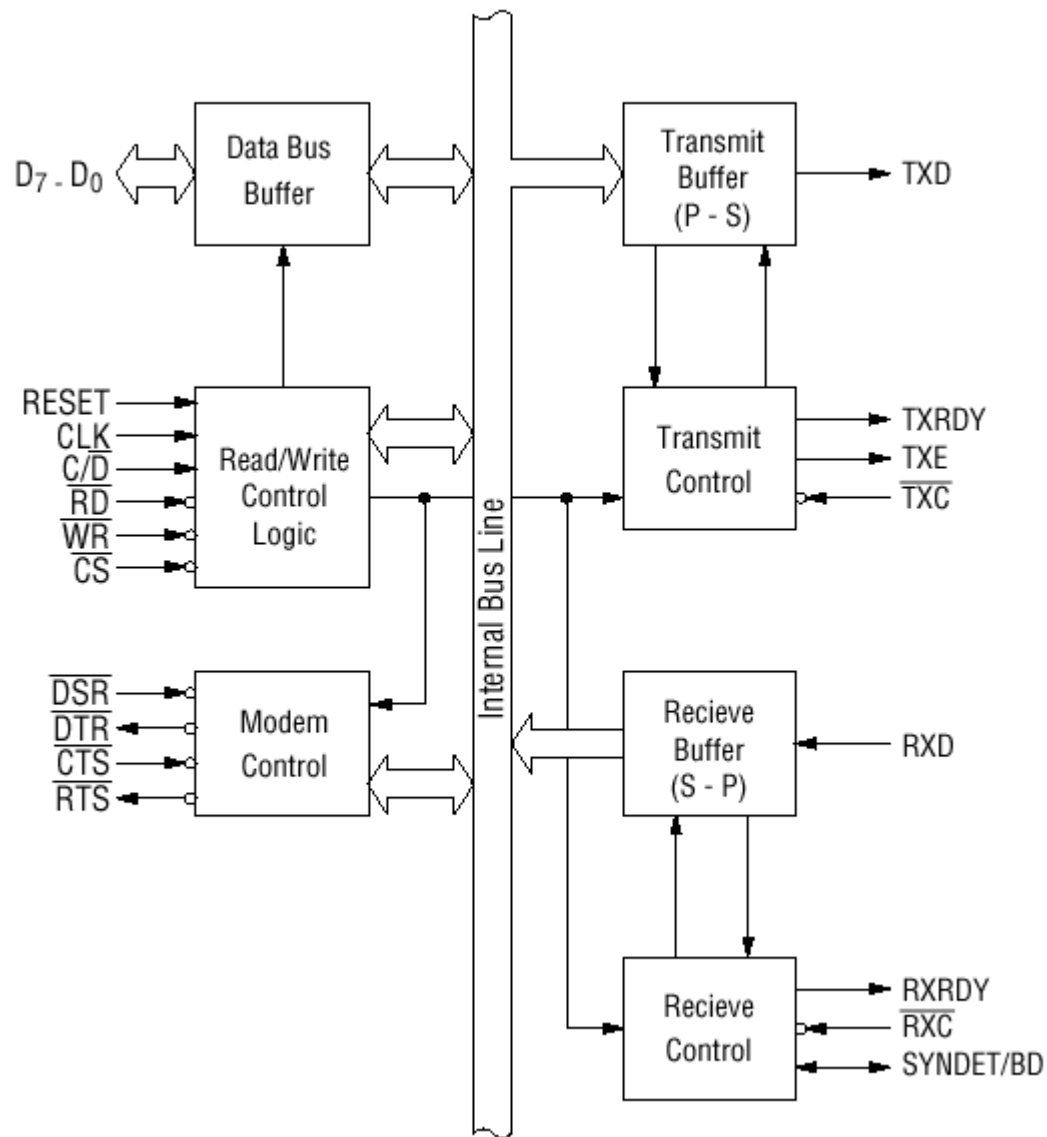
Command Instruction

After the mode instruction, command character should be issued to the USART. It controls the operation of the USART within the basic frame work established by the mode instruction. Fig. 10.7 shows command instruction format.

It does function such as : Enable Transmit/Receive, Error Reset and modem control.

Universal Synchronous Asynchronous Receiver Transmitter (USART)

The 8251 is a USART (Universal Synchronous Asynchronous Receiver Transmitter) for serial data communication. As a peripheral device of a microcomputer system, the 8251 receives parallel data from the CPU and transmits serial data after conversion. This device also receives serial data from the outside and transmits parallel data to the CPU after conversion.



Block diagram of the 8251 USART

The 8251 functional configuration is programed by software. Operation between the 8251 and a CPU is executed by program control. Table 1 shows the operation between a CPU and the device.

\overline{CS}	C/\overline{D}	\overline{RD}	\overline{WR}	
1	×	×	×	Data Bus 3-State
0	×	1	1	Data Bus 3-State
0	1	0	1	Status → CPU
0	1	1	0	Control Word ← CPU
0	0	0	1	Data → CPU
0	0	1	0	Data ← CPU

Table 1 Operation between a CPU and 8251

Control Words

There are two types of control word.

1. Mode instruction (setting of function)
2. Command (setting of operation)

1) Mode Instruction

Mode instruction is used for setting the function of the 8251. Mode instruction will be in "wait for write" at either internal reset or external reset. That is, the writing of a control word after resetting will be recognized as a "mode instruction."

Items set by mode instruction are as follows:

- Synchronous/asynchronous mode
- Stop bit length (asynchronous mode)
- Character length
- Parity bit
- Baud rate factor (asynchronous mode)
- Internal/external synchronization (synchronous mode)
- Number of synchronous characters (Synchronous mode)

The bit configuration of mode instruction is shown in Figures 2 and 3. In the case of synchronous mode, it is necessary to write one-or two byte sync characters. If sync characters were written, a function will be set because the writing of sync characters constitutes part of mode instruction.

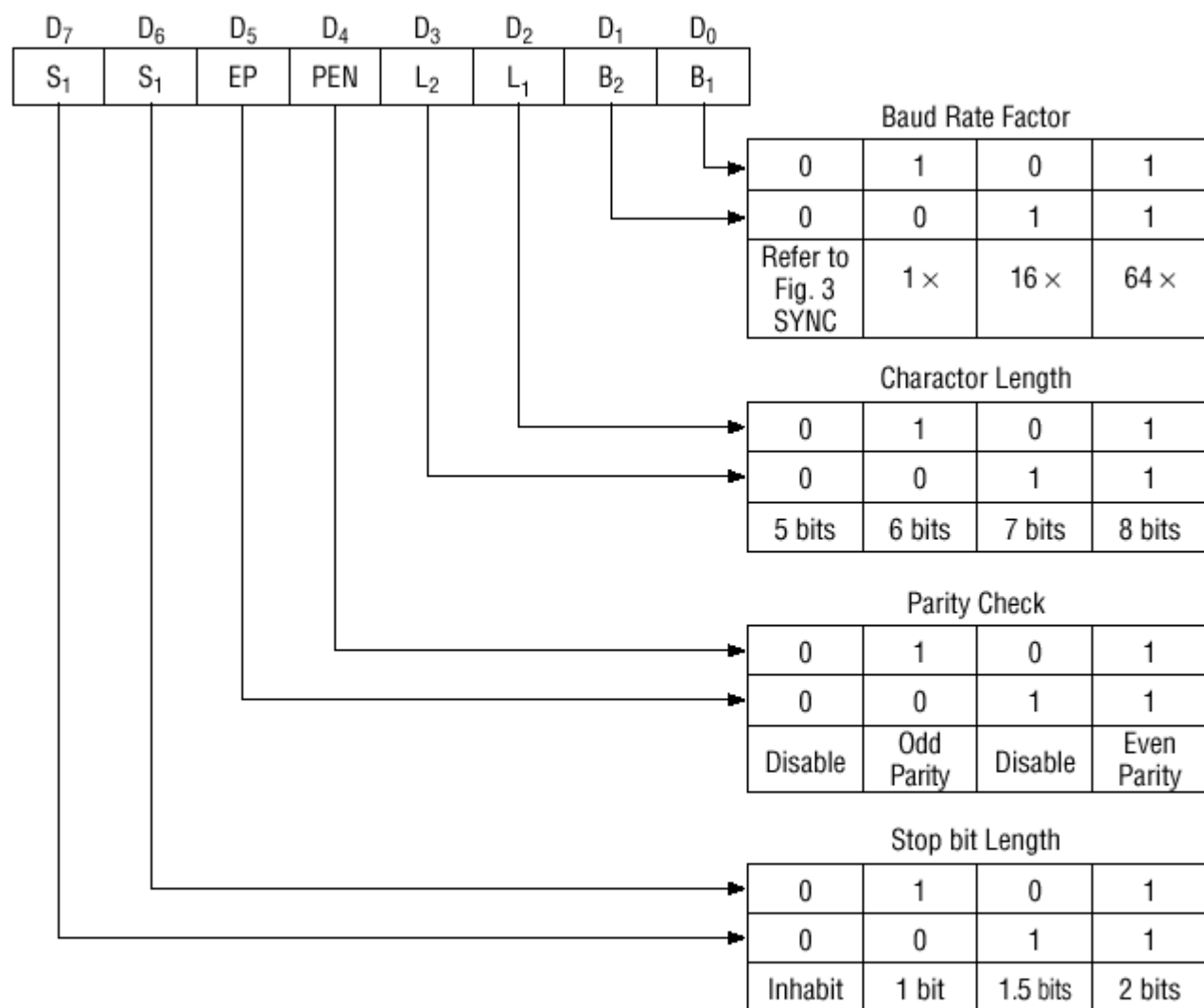


Fig. 2 Bit Configuration of Mode Instruction (Asynchronous)

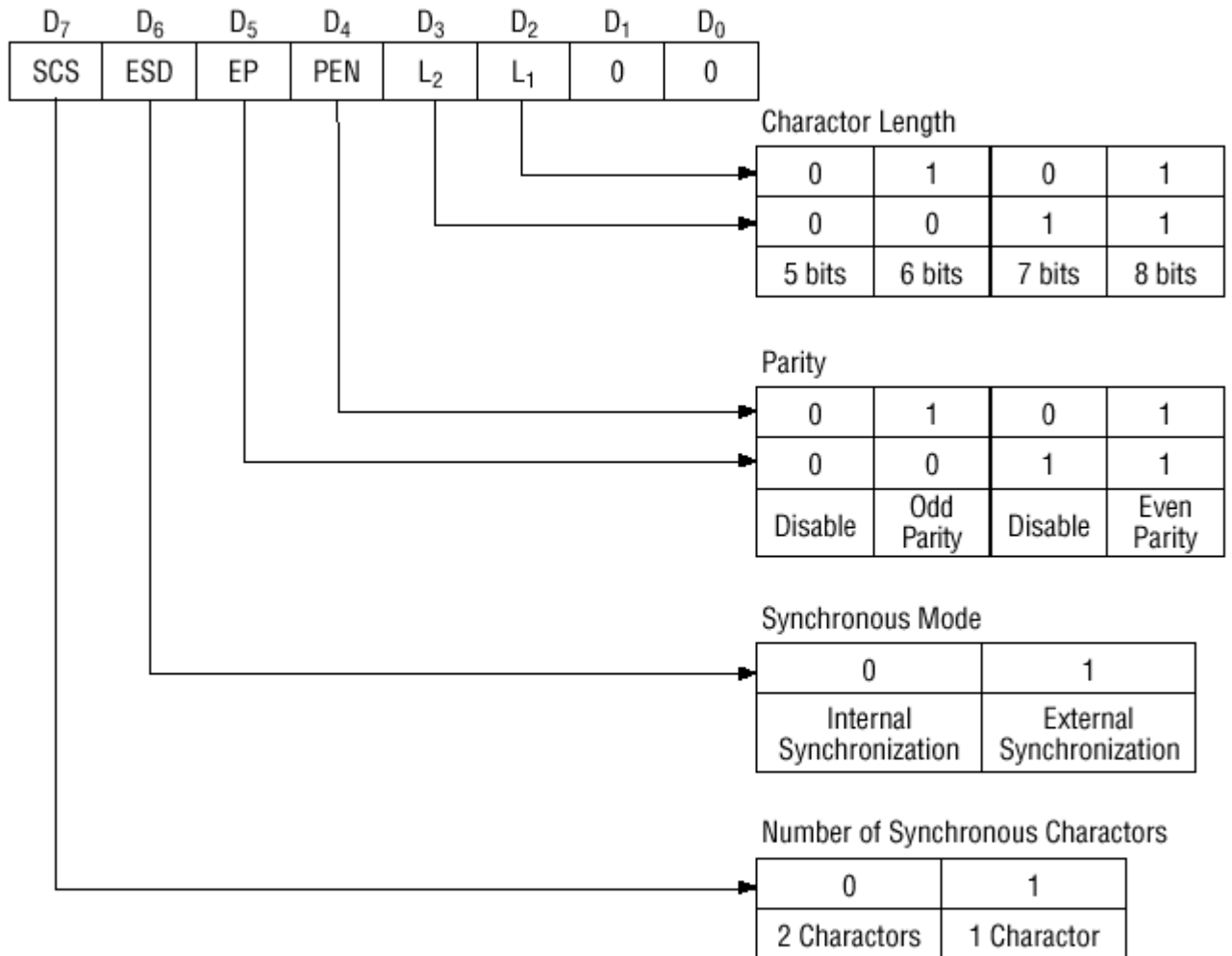


Fig. 3 Bit Configuration of Mode Instruction (Synchronous)

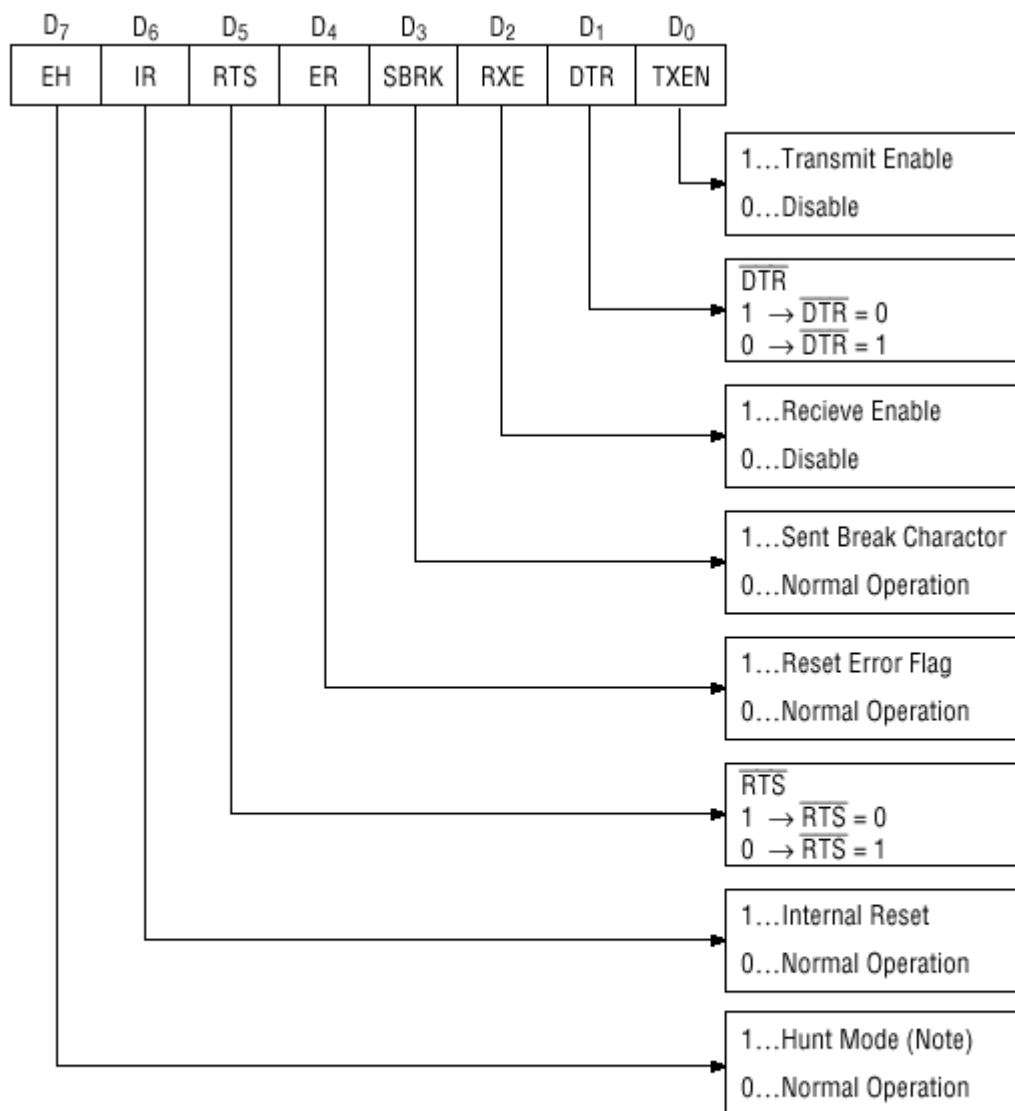
2) Command

Command is used for setting the operation of the 8251. It is possible to write a command whenever necessary after writing a mode instruction and sync characters.

Items to be set by command are as follows:

- Transmit Enable/Disable
- Receive Enable/Disable
- DTR, RTS Output of data.
- Resetting of error flag.
- Sending to break characters
- Internal resetting

- Hunt mode (synchronous mode)



Note: Seach mode for synchronous characters in synchronous mode.

Fig. 4 Bit Configuration of Command

Status Word

It is possible to see the internal status of the 8251 by reading a status word. The bit configuration of status word is shown in Fig. 5.

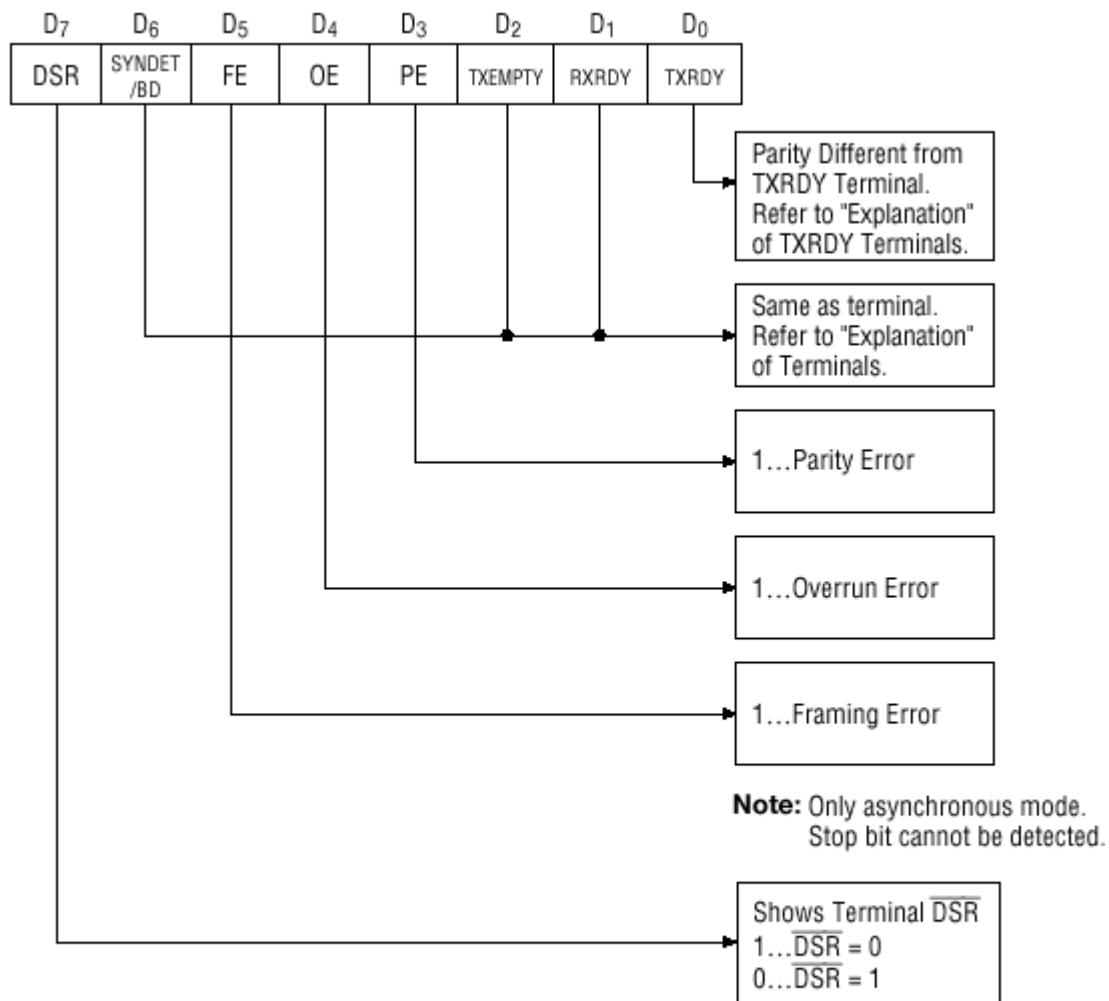


Fig. 5 Bit Configuration of Status Word

Pin Description

D 0 to D 7 (I/O terminal)

This is bidirectional data bus which receive control words and transmits data from the CPU and sends status words and received data to CPU.

RESET (Input terminal)

A "High" on this input forces the 8251 into "reset status." The device waits for the writing of "mode instruction." The min. reset width is six clock inputs during the operating status of CLK.

CLK (Input terminal)

CLK signal is used to generate internal device timing. CLK signal is independent of RXC or TXC. However, the frequency of CLK must be greater than 30 times the RXC and TXC at Synchronous mode and Asynchronous "x1" mode, and must be greater than 5 times at Asynchronous "x16" and "x64" mode.

WR (Input terminal)

This is the "active low" input terminal which receives a signal for writing transmit data and control words from the CPU into the 8251.

RD (Input terminal)

This is the "active low" input terminal which receives a signal for reading receive data and status words from the 8251.

C/D (Input terminal)

This is an input terminal which receives a signal for selecting data or command words and status words when the 8251 is accessed by the CPU. If C/D = low, data will be accessed. If C/D = high, command word or status word will be accessed.

CS (Input terminal)

This is the "active low" input terminal which selects the 8251 at low level when the CPU accesses. Note: The device won't be in "standby status"; only setting CS = High.

TXD (output terminal)

This is an output terminal for transmitting data from which serial-converted data is sent out. The device is in "mark status" (high level) after resetting or during a status when transmit is disabled. It is also possible to set the device in "break status" (low level) by a command.

TXRDY (output terminal)

This is an output terminal which indicates that the 8251 is ready to accept a transmitted data character. But the terminal is always at low level if CTS = high or the device was set in "TX disable status" by a command. Note: TXRDY status word indicates that transmit data character is receivable, regardless of CTS or command. If the CPU writes a data character, TXRDY will be reset by the leading edge of WR signal.

TXEMPTY (Output terminal)

This is an output terminal which indicates that the 8251 has transmitted all the characters and had no data character. In "synchronous mode," the terminal is at high level, if transmit data characters are no longer remaining and sync characters are automatically transmitted. If the CPU writes a data character, TXEMPTY will be reset by the leading edge of WR signal. Note : As the transmitter is disabled by setting CTS "High" or command, data written before disable will be sent out. Then TXD and TXEMPTY will be "High". Even if a data is written after disable, that data is not sent out and TXE will be "High". After the transmitter is enabled, it sent out. (Refer to Timing Chart of Transmitter Control and Flag Timing)

TXC (Input terminal)

This is a clock input signal which determines the transfer speed of transmitted data. In "synchronous mode," the baud rate will be the same as the frequency of TXC. In "asynchronous mode", it is possible to select the baud rate factor by mode instruction. It can be 1, 1/16 or 1/64 the TXC. The falling edge of TXC sifts the serial data out of the 8251.

RXD (input terminal)

This is a terminal which receives serial data.

RXRDY (Output terminal)

This is a terminal which indicates that the 8251 contains a character that is ready to READ. If the CPU reads a data character, RXRDY will be reset by the leading edge of RD signal. Unless the CPU reads a data character before the next one is received completely, the preceding data will be lost. In such a case, an overrun error flag status word will be set.

RXC (Input terminal)

This is a clock input signal which determines the transfer speed of received data. In "synchronous mode," the baud rate is the same as the frequency of RXC. In "asynchronous mode," it is possible to select the baud rate factor by mode instruction. It can be 1, 1/16, 1/64 the RXC.

SYNDET/BD (Input or output terminal)

This is a terminal whose function changes according to mode. In "internal synchronous mode," this terminal is at high level, if sync characters are received and synchronized. If a status word is read, the terminal will be reset. In "external synchronous mode," this is an input terminal. A "High" on this input forces the 8251 to start receiving data characters.

In "asynchronous mode," this is an output terminal which generates "high level" output upon the detection of a "break" character if receiver data contains a "low-level" space between the stop bits of two continuous characters. The terminal will be reset, if RXD is at high level. After Reset is active, the terminal will be output at low level.

DSR (Input terminal)

This is an input port for MODEM interface. The input status of the terminal can be recognized by the CPU reading status words.

DTR (Output terminal)

This is an output port for MODEM interface. It is possible to set the status of DTR by a command.

CTS (Input terminal)

This is an input terminal for MODEM interface which is used for controlling a transmit circuit. The terminal controls data transmission if the device is set in "TX Enable" status by a command. Data is transmittable if the terminal is at low level.

RTS (Output terminal)

This is an output port for MODEM interface. It is possible to set the status RTS by a command.

Data Communication Types

We know that, 8251A is Universal Synchronous, Asynchronous, Receiver, and Transmitter. Therefore communication can take place with four different ways.

1. Asynchronous transmission
2. Asynchronous reception
3. Synchronous transmission
4. Synchronous reception

These communication modes can be enabled by writing proper mode and command instructions. The mode instruction defines the baud rate (in case of asynchronous mode), character length, number of stop bit(s) and parity type. After writing proper mode instruction it is necessary to write appropriate command instruction depending on the communication type.

Asynchronous Transmission

Transmission can be enabled by setting transmission enable bit (bit 0) in the command instruction. When transmitter is enabled and $\overline{CTS} = 0$ the transmitter is ready to transfer data on TxD line.

Operation : When transmitter is ready to transfer data on TxD line, CPU sends data character and it is loaded in the transmit buffer register. The 8251A then automatically adds a start bit (low level) followed by the data bits (least significant bit first), and the programmed number of STOP bit(s) to each character. It also adds parity information prior to STOP bit(s), as defined by the mode instruction. The character is then transmitted as a serial data stream on the TxD output at the falling edge of \overline{TxC} . The rate of transmission is equal to 1, $\frac{1}{6}$ or $\frac{1}{64}$ that of the \overline{TxC} , as defined by the mode instruction. Fig. 10.9 shows the transmitter output in the asynchronous mode.

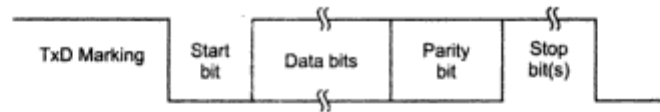


Fig. 10.9 Transmitter output in asynchronous mode

Asynchronous Reception

Reception can be enabled by setting receive enable bit (bit 2) in the command instruction.

Operation :

The RxD line is normally high. 8251A looks for a low level on the RxD line. When it receives the low level, it assumes that it is a START bit and enables an internal counter. At a count equivalent to one-half of a bit time, the RxD line is sampled again. If the line is still low, a valid START bit is detected and the 8251A proceeds to assemble the character. After successful reception of a START bit the 8251A receives data, parity, and STOP bits and then transfers the data on the receiver input register. The data is then transferred into the receiver buffer register. Fig. 10.10 shows the receiver input in the asynchronous mode.

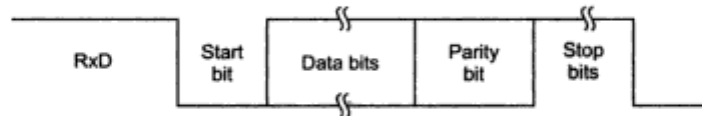


Fig. 10.10 Receiver input in asynchronous mode

Synchronous Transmission

Transmission can be enabled by setting transmission enable bit (bit 0) in the command instruction. When transmitter is enabled and $\overline{\text{CTS}} = 0$, the transmitter is ready to transfer data on TxD line.

Operation : When transmitter is ready to transfer data on TxD line, 8251A transfers characters serially out on the TxD line at the falling edge of the $\overline{\text{TxC}}$. The first character usually is the SYNC character.

Once transmission has started, the data stream at the TxD output must continue at the $\overline{\text{TxC}}$ rate. If CPU does not provide 8251A with a data character before transmitter buffers become empty, the SYNC characters will be automatically inserted in the TxD data stream, as shown in the Fig. 10.11. In this case, the TxEMPTY pin is raised high to indicate CPU that transmitter buffers are empty. The TxEMPTY pin is internally reset when CPU writes data character in the transmitter buffer.

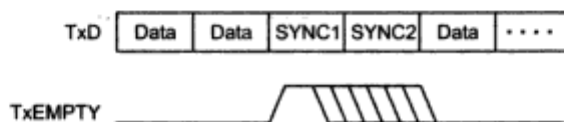


Fig. 10.11 Insertion of SYNC characters

Synchronous Reception : Reception can be enabled by setting receive enable bit (bit 2) in the command instruction.

Operation : In this mode character synchronization can be achieved internally or externally.

Internal SYNC To detect the SYNC character 8251A should be programmed in the 'Enter HUNT' mode by setting bit 7 in the command instruction. Once 8251A enters in the 'Enter HUNT' mode it starts sampling data on the RxD pin on the rising edge of the $\overline{\text{RxC}}$. The content of the receiver buffer is compared at every bit boundary with the first SYNC character until a match occurs. If the 8251A has been programmed for two SYNC characters, the subsequent SYNC characters are compared until the match occurs. Once 8251A detects SYNC character(s) it enters from 'HUNT' mode to character synchronization mode, and starts receiving the data characters on the rising edge of the next $\overline{\text{RxC}}$. To indicate that the synchronization is achieved 8251A sets the SYNDET pin high. It is reset automatically when CPU reads the status register.

External SYNC

In the external SYNC mode, synchronization is achieved by applying a high level on the SYNDET pin, thus forcing the 8251A out of the HUNT mode.

Serial Communication Protocol (RS232C)

In response to the need for signals and handshake standards between DTE and DCE, the Electronic Industries Association (EIA) introduced EIA standard RS-232 in 1962. It was revised and named as RS-232C, in 1969 by EIA. It is widely accepted for single ended data transmission over short distances with low data rates.

This standard describes the functions of 25 signal and handshake pins for serial data transfer. It also describes the voltage levels, impedance levels, rise and fall times, maximum bit rate, and maximum capacitance for these signal lines. RS-232C specifies 25 signal pins and it specifies that the DTE connector should be male, and the DCE connector should be a female. The most commonly used connector should be a female. The most commonly used connector, DB-25P is shown in the Fig. 10.16.

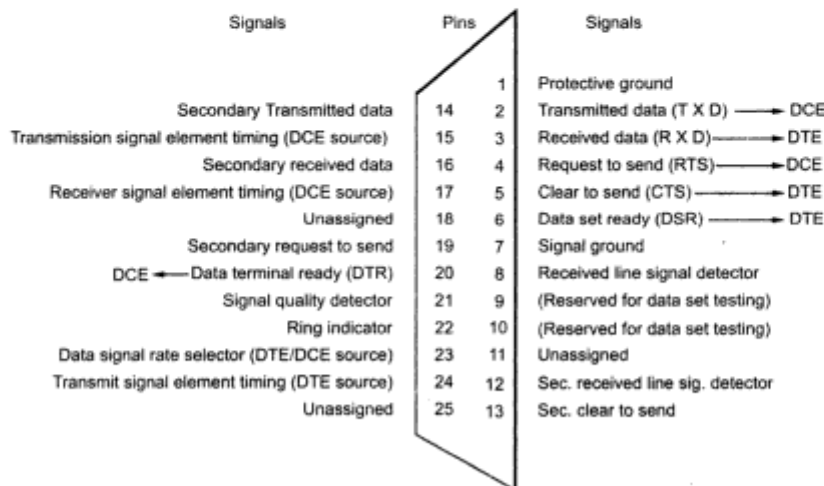


Fig. 10.16 RS 232C 25 pin connector

The Table 10.2 shows pins and signals description for RS-232C for data lines, The voltage level +3V to +15V is defined as logic 0; from -3 V to -15 V is defined as logic 1. The control and timing signals are compatible with the TTL level. Because of the incompatibility of the data lines with the TTL logic, voltage translators, called line drivers and line receivers, are required to interface TTL logic with the RS-232 signals. Fig. 10.17 shows the interfacing between TTL and RS-232 signals. The line driver, MC1488, converts logic 1 into approximately 9 V. These levels at the receiving end are again converted by the line receiver, MC1489, into TTL-compatible logic.

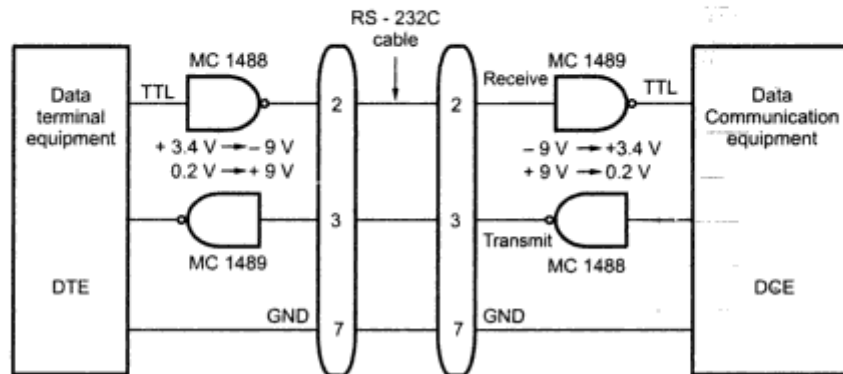


Fig. 10.17 Line drivers and receivers

UNIT-IV

8051 MICROCONTROLLER:

The Intel 8051 microcontroller is one of the most popular general purpose microcontrollers in use today. The success of the Intel 8051 spawned a number of clones which are collectively referred to as the MCS-51 family of microcontrollers, which includes chips from vendors such as Atmel, Philips, Infineon, and Texas Instruments

The Intel 8051 is an 8-bit microcontroller which means that most available operations are limited to 8 bits. There are 3 basic "sizes" of the 8051: Short, Standard, and Extended. The Short and Standard chips are often available in DIP (dual in-line package) form, but the Extended 8051 models often have a different form factor, and are not "drop-in compatible". All these things are called 8051 because they can all be programmed using 8051 assembly language, and they all share certain features (although the different models all have their own special features).

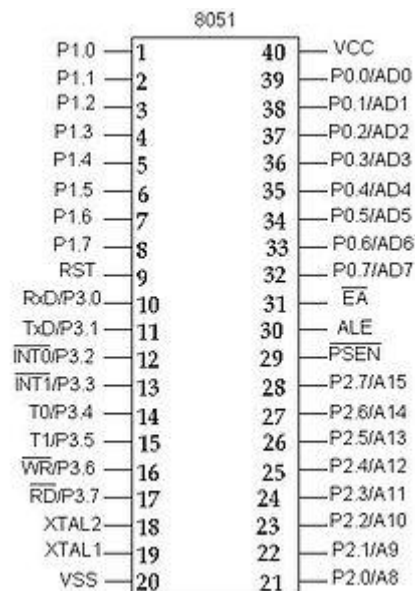
Some of the features that have made the 8051 popular are:

- 64 KB on chip program memory.
- 128 bytes on chip data memory(RAM).
- 4 reg banks.
- 128 user defined software flags.
- 8-bit data bus
- 16-bit address bus
- 32 general purpose registers each of 8 bits
- 16 bit timers (usually 2, but may have more, or less).
- 3 internal and 2 external interrupts.
- Bit as well as byte addressable RAM area of 16 bytes.
- Four 8-bit ports, (short models have two 8-bit ports).
- 16-bit program counter and data pointer.
- 1 Microsecond instruction cycle with 12 MHz Crystal.

8051 models may also have a number of special, model-specific features, such as UARTs, ADC, OpAmps, etc...

Typical applications

8051 chips are used in a wide variety of control systems, telecom applications, robotics as well as in the automotive industry. By some estimations, 8051 family chips make up over 50% of the embedded chip market.



Pin diagram of the 8051 DIP

Basic Pins

PIN 9: PIN 9 is the reset pin which is used reset the microcontroller's internal registers and ports upon starting up. (Pin should be held high for 2 machine cycles.)

PINS 18 & 19: The 8051 has a built-in oscillator amplifier hence we need to only connect a crystal at these pins to provide clock pulses to the circuit.

PIN 40 and 20: Pins 40 and 20 are VCC and ground respectively. The 8051 chip needs +5V 500mA to function properly, although there are lower powered versions like the Atmel 2051 which is a scaled down version of the 8051 which runs on +3V.

PINS 29, 30 & 31: As described in the features of the 8051, this chip contains a built-in flash memory. In order to program this we need to supply a voltage of +12V at pin 31. If external memory is connected then PIN 31, also called EA/VPP, should be connected to ground to indicate the presence of external memory. PIN 30 is called ALE (address latch enable), which is used when multiple memory chips are connected to the controller and only one of them needs to be selected. We will deal with this in depth in the later chapters. PIN 29 is called PSEN. This is "program store enable". In order to use the external memory it is required to provide the low voltage (0) on both PSEN and EA pins.

Ports

There are 4 8-bit ports: P0, P1, P2 and P3.

PORT P1 (Pins 1 to 8): The port P1 is a general purpose input/output port which can be used for a variety of interfacing tasks. The other ports P0, P2 and P3 have dual roles or additional functions associated with them based upon the context of their usage.

PORT P3 (Pins 10 to 17): PORT P3 acts as a normal IO port, but Port P3 has additional functions such as, serial transmit and receive pins, 2 external interrupt pins, 2 external counter inputs, read and write pins for memory access.

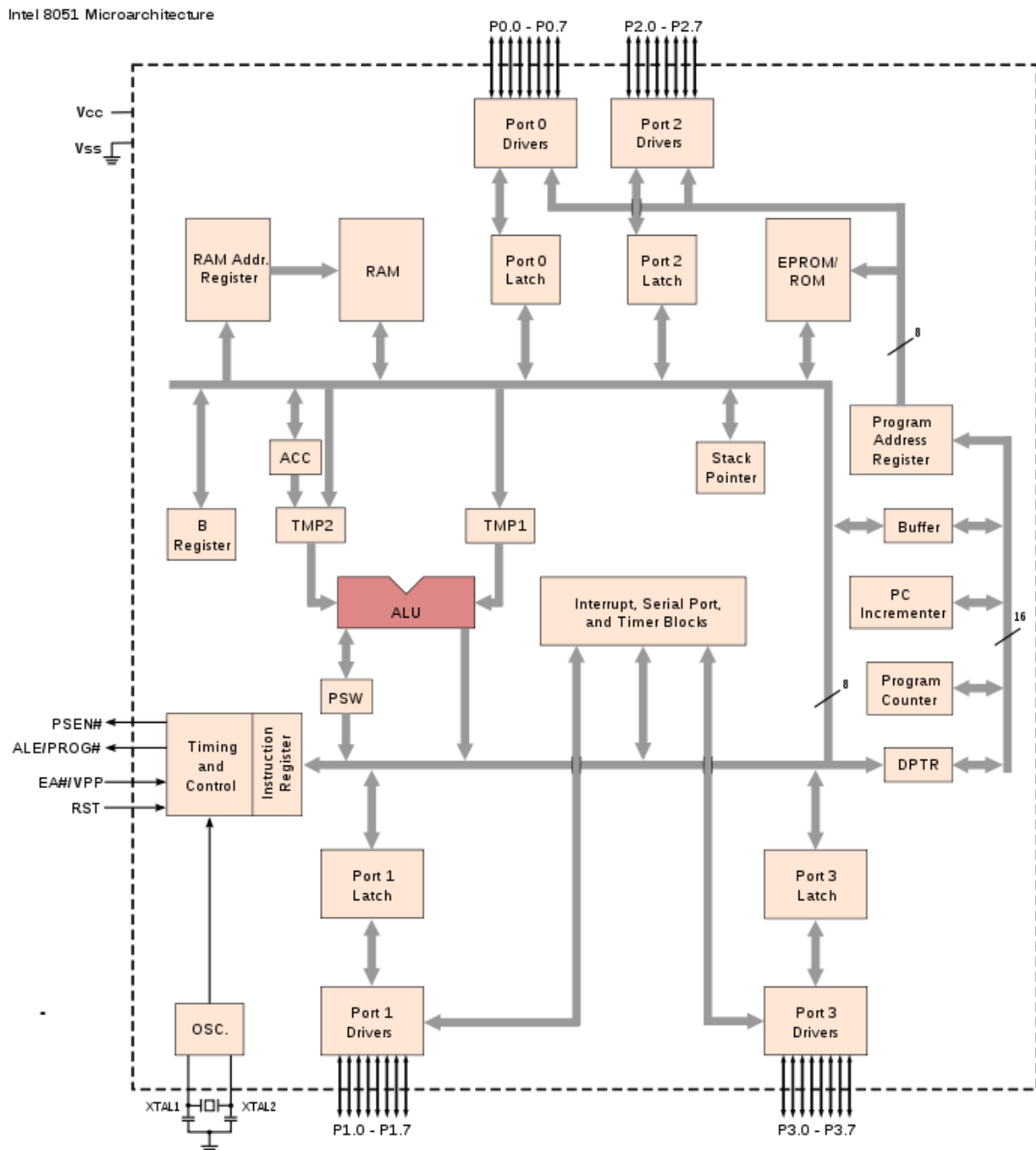
PORT P2 (pins 21 to 28): PORT P2 can also be used as a general purpose 8 bit port when no external memory is present, but if external memory access is required then PORT P2 will act as an address bus in conjunction with PORT P0 to access external memory. PORT P2 acts as A8-A15, as can be seen from fig 1.1

PORT P0 (pins 32 to 39) PORT P0 can be used as a general purpose 8 bit port when no external memory is present, but if external memory access is required then PORT P0 acts as a multiplexed address and data bus that can be used to access external memory in conjunction with PORT P2. P0 acts as AD0-AD7, as can be seen from fig 1.1

Oscillator Circuits

The 8051 requires the existence of an external oscillator circuit. The oscillator circuit usually runs around 12MHz, although the 8051 (depending on which specific model) is capable of running at a maximum of 40MHz. Each machine cycle in the 8051 is 12 clock cycles, giving an effective cycle rate at 1MHz (for a 12MHz clock) to 3.33MHz (for the maximum 40MHz clock).

Internal Architecture



Data and Program Memory

The 8051 Microprocessor can be programmed in PL/M, 8051 Assembly, C and a number of other high-level languages. Many compilers even have support for compiling C++ for an 8051.

Program memory in the 8051 is read-only, while the data memory is considered to be read/write accessible. When stored on EEPROM or Flash, the program memory can be rewritten when the microcontroller is in the special programmer circuit.

Program Start Address

The 8051 starts executing program instructions from address 0000 in the program memory.

Direct Memory

The 8051 has 256 bytes of internal addressable RAM, although only the first 128 bytes are available for general use by the programmer. The first 128 bytes of RAM (from 0x00 to 0x7F) are called the **Direct Memory**, and can be used to store data.

Special Function Register

The **Special Function Register** (SFR) is the upper area of addressable memory, from address 0x80 to 0xFF. A, B, PSW, DPTR are called SFR. This area of memory cannot be used for data or program storage, but is instead a series of memory-mapped ports and registers. All port input and output can therefore be performed by memory **mov** operations on specified addresses in the SFR. Also, different status registers are mapped into the SFR, for use in checking the status of the 8051, and changing some operational parameters of the 8051.

General Purpose Registers

The 8051 has 4 selectable banks of 8 addressable 8-bit registers, R0 to R7. This means that there are essentially 32 available general purpose registers, although only 8 (one bank) can be directly accessed at a time. To access the other banks, we need to change the current bank number in the flag status register.

A and B Registers

The A register is located in the SFR memory location 0xE0. The A register works in a similar fashion to the AX register of x86 processors. The A register is called the **accumulator**, and by default it receives the result of all arithmetic operations. The B register is used in a similar manner, except that it can receive the extended answers from the multiply and divide operations. When not being used for multiplication and Division, the B register is available as an extra general-purpose register.

Comparison between Microprocessor and Microcontroller

We have discussed what is a microprocessor and a microcontroller. Let us see the points of differences between them.

No.	Microprocessor	Microcontroller
1.	Microprocessor contains ALU, control unit (clock and timing circuit), different register and interrupt circuit.	Microcontroller contains microprocessor, memory (ROM and RAM), I/O interfacing circuit and peripheral devices such as A/D converter, serial I/O, timer etc.
2.	It has many instructions to move data between memory and CPU.	It has one or two instructions to move data between memory and CPU.
3.	It has one or two bit handling instructions.	It has many bit handling instructions.
4.	Access times for memory and I/O devices are more.	Less access times for built-in memory and I/O devices.
5.	Microprocessor based system requires more hardware.	Microcontroller based system requires less hardware reducing PCB size and increasing the reliability.
6.	Microprocessor based system is more flexible in design point of view.	Less flexible in design point of view.
7.	It has single memory map for data and code.	It has separate memory map for data and code.
8.	Less number of pins are multifunctioned.	More number pins are multifunctioned.

Features of 8051

The features of the 8051 family are as follows :

- 1) 4096 bytes on - chip program memory.
- 2) 128 bytes on - chip data memory.
- 3) Four register banks.
- 4) 128 User-defined software flags.
- 5) 64 Kilobytes each program and external RAM addressability.
- 6) One microsecond instruction cycle with 12 MHz crystal.
- 7) 32 bidirectional I/O lines organized as four 8-bit ports (16 lines on 8031).
- 8) Multiple mode, high-speed programmable serial port.
- 9) Two multiple mode, 16-bit Timers/Counters.
- 10) Two-level prioritized interrupt structure.
- 11) Full depth stack for subroutine return linkage and data storage.
- 12) Direct Byte and Bit addressability.
- 13) Binary or Decimal arithmetic.
- 14) Signed-overflow detection and parity computation.
- 15) Hardware Multiple and Divide in 4 μ sec.
- 16) Integrated Boolean Processor for control applications.
- 17) Upwardly compatible with existing 8084 software.

8051 Microcontroller Hardware

The Fig. 11.1 shows the internal block diagram of 8051. It consists of a CPU, two kinds of memory sections (data memory - RAM and program memory - EPROM/ROM), input/output ports, special function registers and control logic needed for a variety of peripheral functions. These elements communicate through an eight bit data bus which runs throughout the chip referred as internal data bus. This bus is buffered to the outside world through an I/O port when memory or I/O expansion is desired.

Central Processing Unit (CPU)

The CPU of 8051 consists of eight-bit Arithmetic and Logic unit with associated registers like A, B, PSW, SP, the sixteen bit program counter and "Data pointer" (DPTR) registers. Along with these registers it has a set of special function registers. Along with these registers it has a set of special function registers.

The 8051's ALU can perform arithmetic and logic functions on eight bit variables. The arithmetic unit can perform addition, subtraction, multiplication and division. The logic unit can perform logical operations such as AND, OR, and Exclusive-OR, as well as rotate, clear, and complement. The ALU also looks after the branching decisions. An important and unique feature of the 8051 architecture is that the ALU can also manipulate one bit as well as eight-bit data types. Individual bits may be set, cleared, complemented, moved, tested, and used in logic computation.

Internal RAM

The 8051 has 128-byte internal RAM. It is accessed using RAM address register. The Fig. 11.3 shows the organisation of internal RAM. As shown in the Fig. 11.3, internal RAM of 8051 is organised into three distinct areas :

- Working registers
 - Bit Addressable
 - General Purpose
1. First thirty-two bytes from address 00H to 1FH of internal RAM constitute 32 working registers. They are organised into four banks of eight registers each. The four register banks are numbered 0 to 3 and are consists of eight registers named R_0 to R_7 . Each register can be addressed by name or by its RAM address. Only one register bank is in use at a time. Bits RS_0 and RS_1 in the PSW determine which bank of registers is currently in use. Register banks when not selected can be used as general purpose RAM. On reset, the Bank 0 is selected.
 2. The 8051 provides 16 bytes of a bit-addressable area. It occupies RAM byte addresses from 20H to 2FH, forming a total of 128 (16×8) addressable bits. An addressable bit may be specified by its bit address of 00H to 7FH, or 8 bits may form any byte address from 20H to 2FH. For example, bit address 4EH refers bit 6 of the byte address 29H.
 3. The RAM area above bit addressable area from 30H to 7FH is called general purpose RAM. It is addressable as byte.

See Fig. 11.3 on next page.

11.3.4 Internal ROM

The 8051 has 4 Kbyte of internal ROM with address space from 0000H to 0FFFH. It is programmed by manufacturer when the chip is built. This part cannot be erased or altered after fabrication. This is used to store final version of the program.

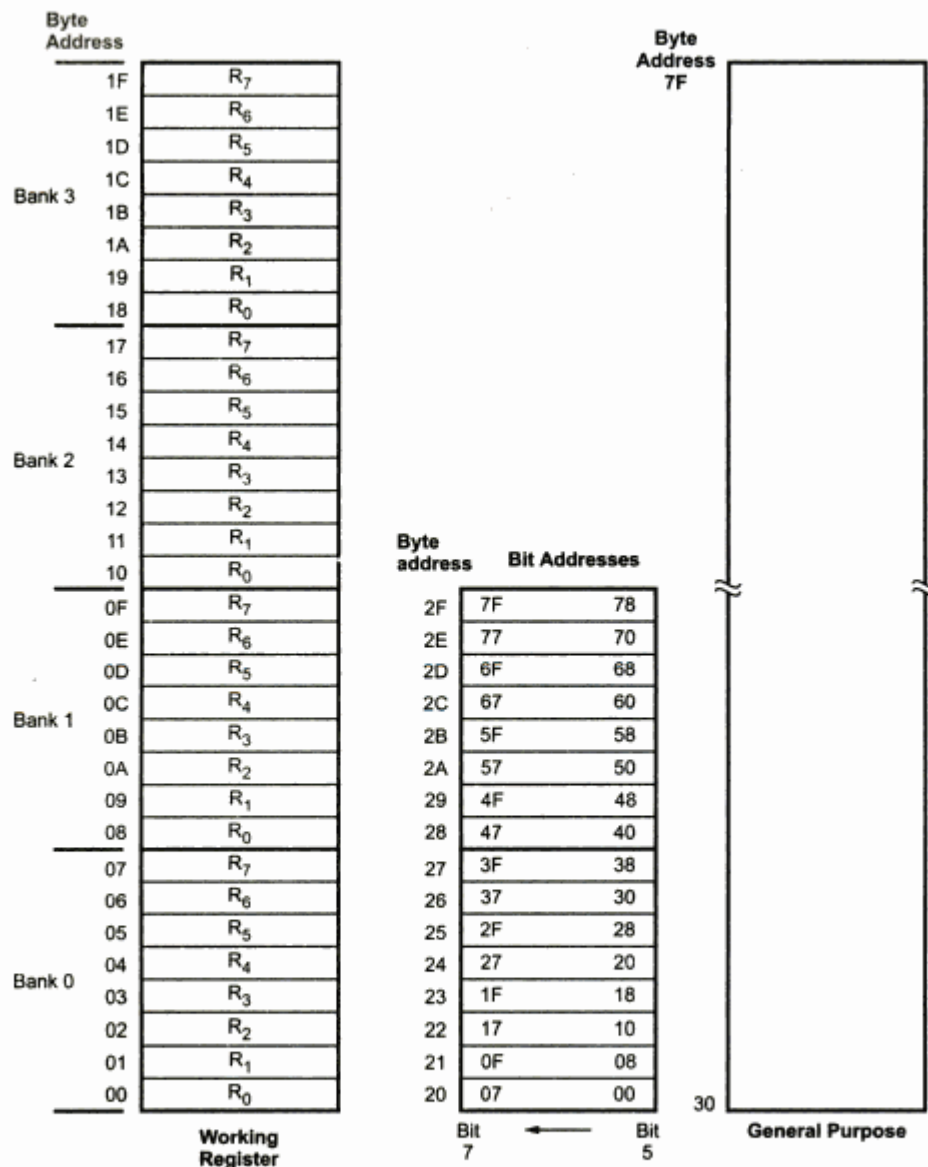


Fig. 11.3 Organisation of internal RAM of 8051

Input/Output Ports

The 8051 has 32 I/O pins configured as four eight-bit parallel ports (P0, P1, P2, and P3). All four ports are bidirectional, i.e. each pin will be configured as input or output (or both) under software control. Each port consists of a latch, an output driver, and an input buffer.

The output drives of Ports 0 and 2 and the input buffers of Port 0, are used to access external memory. As mentioned earlier, Port 0 outputs the low order byte of the external memory address, time multiplexed with the data being written or read, and Port 2 outputs the high order byte of the external memory address when the address is 16 bits wide. Otherwise Port 2 gives the contents of special function register P2.

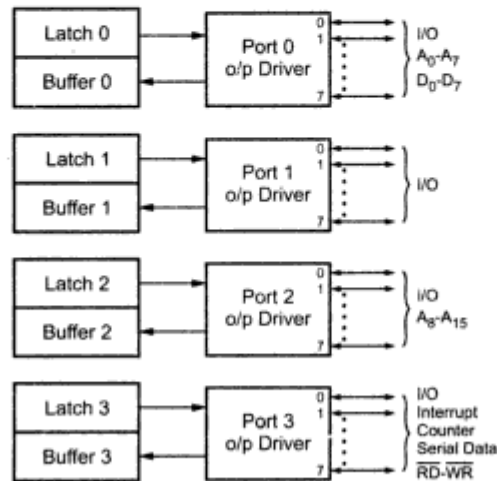


Fig. 11.4 I/O Ports

11.3.6 Register Set of 8051

11.3.6.1 Register A (Accumulator)

It is an 8-bit register. It holds a source operand and receives the result of the arithmetic instructions (addition, subtraction, multiplication, and division). The accumulator can be the source or destination for logical operations and a number of special data movement instructions, including look-up tables and external RAM expansion. Several functions apply exclusively to the accumulator : rotate, parity computation , testing for zero , and so on.

11.3.6.2 Register B

In addition to accumulator, an 8-bit B-register is available as a general purpose register when it is not being used for the hardware multiply/divide operation.

11.3.6.3 Program Status Word (Flag Register)

Many instructions implicitly or explicitly affect (or are affected by) several status flags, which are grouped together to form the Program Status Word. Fig. 11.5 shows the bit pattern of the program status word. It is an 8-bit word, containing the information as follows.

	B ₇	B ₆	B ₅	B ₄	B ₃	B ₂	B ₁	B ₀
	CY	AC	F0	RS1	RS0	OV	-	P

CY Bit 7 - Carry flag
 AC Bit 6 - Auxiliary carry flag for BCD operations
 F0 Bit 5 - User defined flag (Flag zero)
 RS1, RS0 Bit 4-3 - Select the working register banks as follows :

RS1	RS0	Bank Selection	
0	0	00H - 07H	Bank 0
0	1	08H - 0FH	Bank 1
1	0	10H - 17H	Bank 2
1	1	18H - 1FH	Bank 3

OV Bit 2 - Overflow flag
 - Bit 1 - Reserved
 P Bit 0 - Parity flag (1 = Even parity)

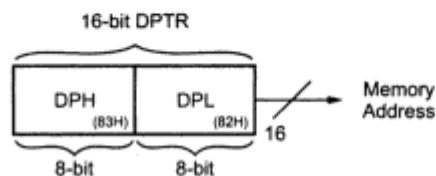
Fig. 11.5 Program status word

11.3.6.4 Stack and Stack Pointer

The stack refers to an area of internal RAM that is used in conjunction with certain opcodes data to store and retrieve data quickly. The stack pointer register is used by the 8051 to hold an internal RAM address that is called **top of stack**. The stack pointer register is 8-bit wide. It is increased **before** data is stored during PUSH and CALL instructions and decremented **after** data is restored during POP and RET instructions. Thus stack array can reside anywhere in on-chip RAM. The stack pointer is initialized to 07H after a reset. This causes the stack to begin at location 08H. The operation of stack and stack pointer is illustrated in Fig. 11.6.

11.3.6.5 Data Pointer (DPTR)

The data pointer (DPTR) consists of a high byte (DPH) and a low byte (DPL). Its function is to hold a 16 bit address. It may be manipulated as a 16 bit data register or as two independent 8 bit registers. It serves as a base register in indirect jumps, lookup table instructions and external data transfer. The DPTR does not have a single internal address; DPH (83H) and DPL (82H) have separate internal addresses.



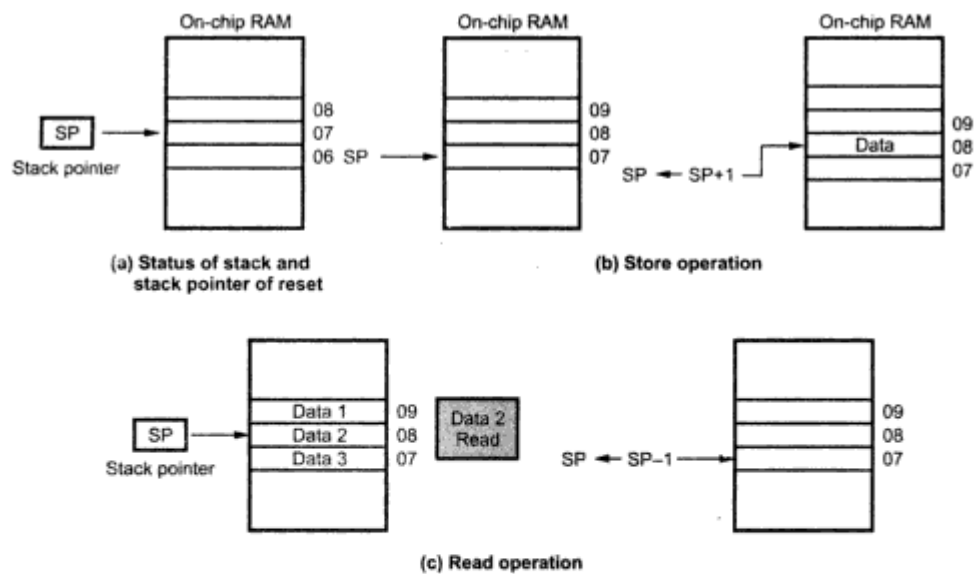


Fig. 11.6

11.3.6.6 Program Counter

The 8051 has a 16-bit program counter. It is used to hold the address of memory location from which the next instruction is to be fetched. Due to this the width of the program counter decides the maximum program length in bytes. For example, 8051 is 16-bit hence it can address upto 2^{16} bytes (64 K) of memory.

The PC is automatically incremented to point the next instruction in the program sequence after execution of the current instruction. It may also be altered by certain instructions. The PC is the only register that does not have an internal address.

11.3.6.7 Special Function Registers

Unlike other microprocessors in the Intel family, 8051 uses memory mapped I/O through a set of special function registers that are implemented in the address space immediately above the 128 bytes of RAM. Fig. 11.7 shows special function bit addresses. All access to the four I/O ports, the CPU registers, interrupt-control registers, the timer/counter, UART, and power control are performed through registers between 80H and FFH.

Direct Byte Address (MSB)	Bit Address (LSB)								Hardware Register Symbol
0FFH									
0F0H	F7	F6	F5	F4	F3	F2	F1	F0	B
0E0H	E7	E6	E5	E4	E3	E2	E1	E0	ACC
0D0H	D7	D6	D5	D4	D3	D2	D1	D0	PSW
0B8H	---	---	---	BC	BB	BA	B9	B8	IP
0B0H	B7	B6	B5	B4	B3	B2	B1	B0	P3
0A8H	AF	---	---	AC	AB	AA	A9	A8	IE
0A0H	A7	A6	A5	A4	A3	A2	A1	A0	P2
98H	9F	9E	9D	9C	9B	9A	99	98	SCON
90H	97	96	95	94	93	92	91	90	P1
88H	8F	8E	8D	8C	8B	8A	89	88	TCON
80H	87	86	85	84	83	82	81	80	P0

Fig. 11.7 SFR bit address

Symbol	Name	Address	Value in Binary
*ACC	Accumulator	0E0H	0 0 0 0 0 0 0 0
*B	B Register	0F0H	0 0 0 0 0 0 0 0
*PSW	Program Status Word	0D0H	0 0 0 0 0 0 0 0
SP	Stack Pointer	81H	0 0 0 0 0 1 1 1
DPTR	Data Pointer 2 Bytes		
DPL	Low Byte	82H	0 0 0 0 0 0 0 0
DPH	High Byte	83H	0 0 0 0 0 0 0 0
*P0	Port 0	80H	1 1 1 1 1 1 1 1
*P1	Port 1	90H	1 1 1 1 1 1 1 1
*P2	Port 2	0A0H	1 1 1 1 1 1 1 1
*P3	Port 3	0B0H	1 1 1 1 1 1 1 1
*IP	Interrupt Priority Control	0B8H	8051 X X X 0 0 0 0 0 8052 X X 0 0 0 0 0 0
*IE	Interrupt Enable Control	0A8H	8051 0 X X 0 0 0 0 0 8052 0 X 0 0 0 0 0 0
TMOD	Timer/Counter Mode Control	89H	0 0 0 0 0 0 0 0
*TCON	Timer/Counter Control	88H	0 0 0 0 0 0 0 0
* + T2CON	Timer/Counter 2 Control	0C8H	0 0 0 0 0 0 0 0
TH0	Timer/Counter 0 High Byte	8CH	0 0 0 0 0 0 0 0
TL0	Timer/Counter 0 Low Byte	8AH	0 0 0 0 0 0 0 0
TH1	Timer/Counter 1 High Byte	8DH	0 0 0 0 0 0 0 0
TL1	Timer/Counter 1 LowByte	8BH	0 0 0 0 0 0 0 0
+ TH2	Timer/Counter 2 High Byte	0CDH	0 0 0 0 0 0 0 0
+ TL2	Timer/Counter 2 Low Byte	0CCH	0 0 0 0 0 0 0 0
+ RCAP2H	T/C 2 Capture Reg. High Byte	0CBH	0 0 0 0 0 0 0 0
+ RCAP2L	T/C 2 Capture Reg. Low Byte	0CAH	0 0 0 0 0 0 0 0
* SCON	Serial Control	98H	0 0 0 0 0 0 0 0
SBUF	Serial Data Buffer	99H	Interminate
PCON	Power Control	87H	HMOS 0 X X X X X X X CHMOS 0 X X X 0 0 0 0

Table 11.3 List of all SFRs (* = Bit addressable, + = 8052 only)

Memory Organization in 8051

Fig. 11.8 shows the basic memory structure for 8051. It can access upto 64 K program memory and 64 K data memory. The 8051 has 4 Kbytes of internal program memory and 256 bytes of internal data memory.

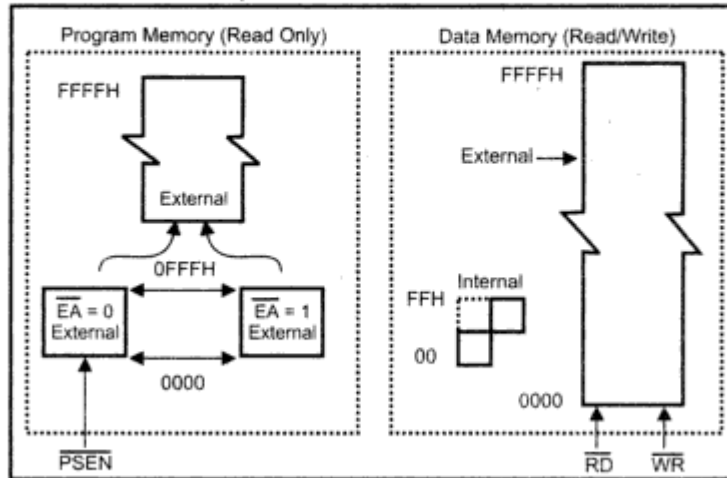


Fig. 11.8 Memory structure

External Data Memory and Program Memory

We have seen that 8051 has internal data and code memory with limited memory capacity. This memory capacity may not be sufficient for some applications. In such situations, we have to connect external ROM/EPROM and RAM to 8051 microcontroller to increase the memory capacity. We also know that ROM is used as a program memory and RAM is used as a data memory. Let us see how 8051 accesses these memories.

External Program Memory

Fig. 11.10 shows a map of the 8051 program memory.

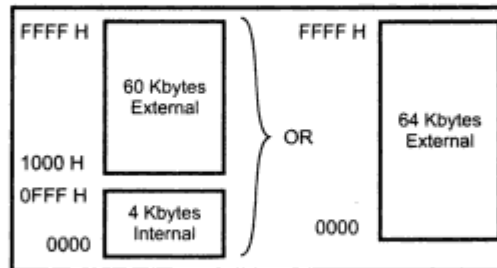


Fig. 11.10 The 8051 program memory

In 8051, when the \overline{EA} pin is connected to V_{CC} , program fetches to addresses 0000H through 0FFFH are directed to the internal ROM and program fetches to addresses 1000H through FFFFH are directed to external ROM/EPROM. On the other hand when \overline{EA} pin is grounded, all addresses (0000H to FFFFH) fetched by program are directed to the external ROM/EPROM. The \overline{PSEN} signal is used to activate output enable signal of the external ROM/EPROM, as shown in the Fig. 11.11.

External Data Memory and Program Memory

We have seen that 8051 has internal data and code memory with limited memory capacity. This memory capacity may not be sufficient for some applications. In such situations, we have to connect external ROM/EPROM and RAM to 8051 microcontroller to increase the memory capacity. We also know that ROM is used as a program memory and RAM is used as a data memory. Let us see how 8051 accesses these memories.

External Program Memory

Fig. 11.10 shows a map of the 8051 program memory.

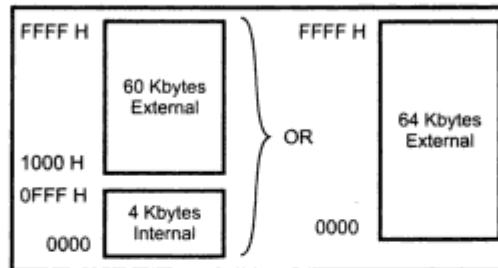


Fig. 11.10 The 8051 program memory

In 8051, when the \overline{EA} pin is connected to V_{CC} , program fetches to addresses 0000H through 0FFFH are directed to the internal ROM and program fetches to addresses 1000H through FFFFH are directed to external ROM/EPROM. On the other hand when \overline{EA} pin is grounded, all addresses (0000H to FFFFH) fetched by program are directed to the external ROM/EPROM. The \overline{PSEN} signal is used to activate output enable signal of the external ROM/EPROM, as shown in the Fig. 11.11.

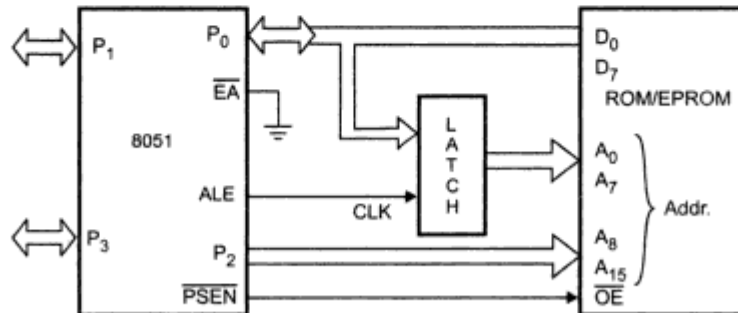


Fig. 11.11 Accessing external program memory

UNIT-V

Timer 0 and Timer 1

In these timers, "Timer" or "Counter" mode is selected by control bits C/\bar{T} in the Special Function Register TMOD (Fig. 11.18). These two Timer/Counters have four operating modes, which are selected by bit-pairs (M1, M0) in TMOD. Modes 0, 1 and 2 are same for both Timer/Counters. Mode 3 is different. The four operating modes are described as follows :

(MSB)				(LSB)			
GATE	C/\bar{T}	M1	M0	GATE	C/\bar{T}	M1	M0
Timer 1				Timer 0			
GATE				M1	M0	Operating Mode	
Gating control when set. Timer/Counter "x" is enabled only while "INTx" pin is high and "TRx" control bit is set. When cleared Timer "x" is enabled whenever "TRx" control bit is set.				0	0	8-bit Timer/Counter "THx" with "TLx's 5-bit prescaler.	
C/\bar{T}				0	1	16-bit Timer/Counter "THx" with "TLx" are cascaded; there is no prescaler.	
Timer or Counter selector cleared for Timer operation (input from internal system clock). Set for Counter operation (input from "Tx" input pin).				1	0	8-bit auto-reload Timer/Counter "THx" holds a value which is to be reloaded into "TLx" each time it overflows.	
				1	1	(Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit timer only controlled by Timer 1 control bits.	
				1	1	(Timer 1) Timer/Counter 1 stopped.	

TMOD : Timer/counter mode control register

MODE 0

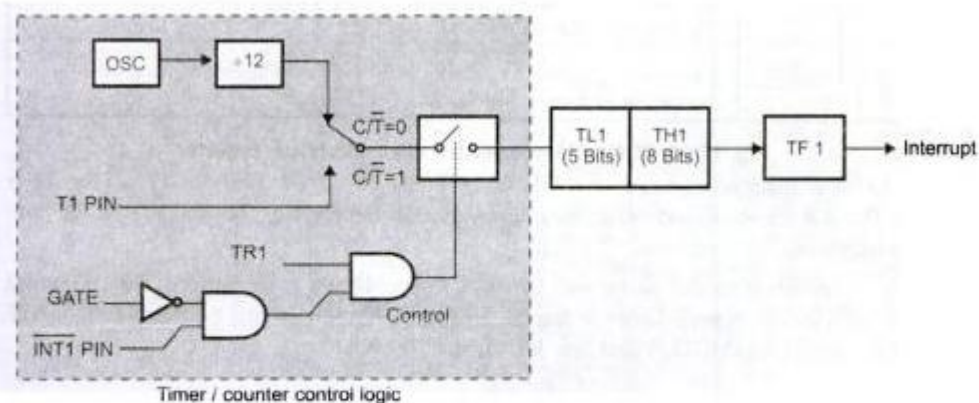


Fig. 11.19 Timer/counter 1 mode 0 : 13-bit counter

Both Timers in Mode 0 is an 8-bit Counter with a divide-by-32 prescaler. This 13-bit timer is MCS-48 compatible. Fig. 11.19 shows the Mode 0 operation as it applies to Timer 1. In this mode, the Timer register is configured as a 13-bit register. As the count rolls over from all 1s to all 0s, it sets the Timer interrupt flag $TF1$. The counted input is enabled to the Timer when $TR1 = 1$ and either $GATE = 0$ or $\overline{INT1} = 1$. (Setting $GATE = 1$ allows the Timer to be controlled by external input $\overline{INT1}$, to facilitate pulse width measurements.) $TR1$ is a control bit in the Special Function Register TCON (Fig. 11.20) $GATE$ is in TMOD.

(MSB)				(LSB)			
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Symbol	Position	Name and Significance
TF1	TCON.7	Timer 1 Overflow Flag. Set by hardware on timer/counter overflow. Cleared when interrupt processed.
TR1	TCON.6	Timer 1 Run control bit. Set/cleared by software to turn timer/counter on/off.
TF0	TCON.5	Timer 0 Overflow Flag. Set by hardware on timer/counter overflow. Cleared when interrupt processed.
TR0	TCON.4	Timer 0 Run control bit. Set/cleared by software to turn timer/counter on/off.
IE1	TCON.3	Interrupt 1 Edge Flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
IT1	TCON.2	Interrupt 1 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.
IE0	TCON.1	Interrupt 0 Edge Flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
IT0	TCON.0	Interrupt 0 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.

Fig 11.20 TCON-timer/counter control/status register

The 13-bit register consists of all 8 bits of TH1 and the lower 5 bits of TL1. The upper 3 bits of TL1 are indeterminate and should be ignored. Setting the run flag ($TR1$) does not clear the registers.

RI	SCON.0	Receive Interrupt flag. Set by hardware when byte received. Cleared by software after servicing.
	Note : Mode	The state of (SM0, SM1) selects ; SM0 SM1
	0	0 0 - Shift register ; baud = $f/12$
	1	0 1 - 8-bit UART, variable data rate.
	2	1 0 - 9-bit UART, fixed data rate ; baud = $f/32$ or $f/64$
	3	1 1 - 9-bit UART, variable data rate.

Fig. 11.24 (a) SCON-serial port control/status register

(MSB)	7	6	5	4	3	2	1	(LSB)
	SMOD	–	–	–	GF1	GF0	PD	IDL

Symbol	Position	Name and Significance
SMOD	PCON.7	Serial baud rate modify bit. It is 0 at reset. It is set to 1 by program to double the baud rate.
–	PCON.6-4	Not defined
GF1	PCON.3	General purpose user flag bit 1. Set/cleared by program.
GF0	PCON.2	General purpose user flag bit 0. Set/cleared by program.
PD	PCON.1	Power down bit. It is set to 1 by program to enter power down configuration for CHMOS microcontrollers.
IDL	PCON.0	Idle mode bit. It is set to 1 by program to enter idle mode configuration for CHMOS microcontrollers.
Note : PCON is not bit addressable		

Fig. 11.24 (b) PCON register

Operating Modes for Serial Port

MODE 0

In this mode, serial data enters and exits through RXD. TXD outputs the shift clock. 8 bits are transmitted/received : 8 data bits (LSB first). The baud rate is fixed at 1/12 the oscillator frequency.

MODE 1

In this mode, 10 bits are transmitted (through TXD) or received (through RXD) : a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in Special Function Register SCON. The baud rate is variable.

MODE 2

In this mode, 11 bits are transmitted (through TXD) or received (through RXD) : a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On Transmit, the 9th data bit (TB8 in SCON) can be assigned the value of 0 or 1. Or, for example, the parity bit (P, in the PSW) could be moved into TB8. On receive, the 9th data bit goes into RB8 in Special Function Register SCON, while the stop bit is ignored. The baud rate is programmable to either $\frac{1}{32}$ or $\frac{1}{64}$ the oscillator frequency.

MODE 3

In this mode, 11 bits are transmitted (through TXD) or received (through RXD) : a start bit (0), 8 data bits (LSB first), a programmable 9th data bit and a stop bit (1). In fact, Mode 3 is the same as Mode 2 in all respects except the baud rate. The baud rate in Mode 3 is variable.

In all four modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1.

The Table 11.9 summaries the four serial port modes provided by 8051.

Mode	Transmission Format	Baud Rate
0	8-data bits	$\frac{1}{12}$ oscillator frequency
1	10-bit (start bit + 8-data bits + stop bit)	Variable
2	11-bit (start bit + 8-data bits + programmable 9 th data bit + stop bit)	Programmable to either $\frac{1}{32}$ or $\frac{1}{64}$ oscillator frequency
3	11-bit (start bit + 8 data bit + programmable 9 th data bit + stop bit)	Variable

Table 11.9 Summary of serial port modes

ADDITIONAL

15.ADDITIONAL TOPICS:

INTRODUCTION TO PENTIUM AND DUAL CORE PROCESSORS

DUAL CORE PROCESSORS

A multi-core processor is a single [computing](#) component with two or more independent actual processors (called "cores"), which are the units that read and execute [program instructions](#).^[1] The instructions are ordinary [CPU instructions](#) such as add, move data, and branch, but the multiple cores can run multiple instructions at the same time, increasing overall speed for programs amenable to parallel. Manufacturers typically integrate the cores onto a single [integrated circuit die](#) (known as a chip multiprocessor or CMP), or onto multiple dies in a single [chip package](#).

In which the number of cores is large enough that traditional multi-processor techniques are no longer efficient—largely because of issues with congestion in supplying instructions and data to the many processors. Processors were originally developed with only one core. A **many-core** processor is a multi-core processor. The many-core threshold is roughly in the range of several tens of cores; above this threshold [network on chip](#) technology is advantageous. [Tilera](#) processors feature a switch in each core to route data through an on-chip mesh network to lessen the data congestion, enabling their core count to scale up to 100 cores.

A **dual-core processor** has two cores (e.g. AMD Phenom II X2, Intel Core Duo), a **quad-core processor** contains four cores (e.g. AMD Phenom II X4, Intel's quad-core processors, see [i3](#), [i5](#), and [i7](#) at [Intel Core](#)), a **hexa-core processor** contains six cores (e.g. [AMD Phenom II X6](#), Intel Core i7 Extreme Edition 980X), an **octa-core processor** contains eight cores (e.g. [Intel Xeon E7-2820](#), AMD FX-8150). A multi-core processor implements [multiprocessing](#) in a single physical package. Designers may couple cores in a multi-core device tightly or loosely. For example, cores may or may not share [caches](#), and they may implement [message passing](#) or [shared memory](#) inter-core communication methods. Common [network topologies](#) to interconnect cores include [bus](#), ring, two-dimensional mesh, and [crossbar](#). *Homogeneous* multi-core systems include only identical cores, *heterogeneous* multi-core systems have cores that are not identical. Just as with single-processor systems, cores in multi-core systems may implement architectures such as [superscalar](#), [VLIW](#), [vector processing](#), [SIMD](#), or [multithreading](#).

Multi-core processors are widely used across many application domains including general-purpose, [embedded](#), [network](#), [digital signal processing](#) (DSP), and [graphics](#).

Terminology

The terms *multi-core* and *dual-core* most commonly refer to some sort of central processing unit (CPU), but are sometimes also applied to digital signal processors (DSP) and system-on-a-chip (SoC circuit, unless otherwise noted.). The terms are generally used only to refer to multi-core microprocessors that are manufactured on the *same* integrated circuit die; separate microprocessor dies in the same package are generally referred to by another name, such as *multi-chip module*. This article uses the terms "multi-core" and "dual-core" for CPUs manufactured on the *same* integrated circuits.

Commercial incentives

Several business motives drive the development of dual-core architectures. For decades, it was possible to improve performance of a CPU by shrinking the area of the integrated circuit, which drove down the cost per device on the IC. Alternatively, for the same circuit area, more transistors could be utilized in the design, which increased functionality, especially for CISC architectures. Clock rates also increased by orders of magnitude in the decades of the late 20th century, from several megahertz in the 1980s to several gigahertz in the early 2000s.

As the rate of clock speed improvements slowed, increased use of parallel computing in the form of multi-core processors has been pursued to improve overall processing performance. Multiple cores were used on the same CPU chip, which could then lead to better sales of CPU chips with two or more cores. Intel has produced a 48-core processor for research in cloud computing

Technical factors

Since computer manufacturers have long implemented symmetric multiprocessing (SMP) designs using discrete CPUs, the issues regarding implementing multi-core processor architecture and supporting it with software are well known.

Additionally:

Utilizing a proven processing-core design without architectural changes reduces design risk significantly.

- For general-purpose processors, much of the motivation for multi-core processors comes from greatly diminished gains in processor performance from increasing the operating frequency. This is due to three primary factors:
 1. The *memory wall*; the increasing gap between processor and memory speeds. This effect pushes cache sizes larger in order to mask the latency of memory. This helps only to the extent that memory bandwidth is not the bottleneck in performance.
 2. The *ILP wall*; the increasing difficulty of finding enough parallelism in a single instructions stream to keep a high-performance single-core processor busy.
 3. The *power wall*; the trend of consuming exponentially increasing power with each factorial increase of operating frequency. This increase can be mitigated by "shrinking" the processor by using smaller traces for the same logic. The *power wall* poses manufacturing, system design and deployment problems that have not been justified in the face of the diminished gains in performance due to the *memory wall* and *ILP wall*.

In order to continue delivering regular performance improvements for general-purpose processors, manufacturers such as Intel and AMD have turned to multi-core designs, sacrificing lower manufacturing-costs for higher performance in some applications and systems. Multi-core architectures are being developed, but so are the alternatives. An especially strong contender for established markets is the further integration of peripheral functions into the chip.

The **Pentium Dual-Core** brand was used for mainstream x86 architecture microprocessors from Intel from 2006 to 2009 when it was renamed to Pentium. The processors are based on either the 32-bit *Yonah* or (with quite different microarchitectures) 64-bit *Merom-2M*, *Allendale*, and *Wolfdale-3M* core, targeted at mobile or desktop computers.

In terms of features, price and performance at a given clock frequency, Pentium Dual-Core processors were positioned above Celeron but below Core and Core 2 microprocessors in Intel's product range. The Pentium Dual-Core was also a very popular choice for overclocking, as it can deliver high performance (when overclocked) at a low price.

Processor cores

In 2006, Intel announced a plan^[1] to return the Pentium trademark from retirement to the market, as a moniker of low-cost Core microarchitecture processors based on the single-core Conroe-L but with 1 MiB of cache. The identification numbers for those planned Pentiums were similar to the numbers of the latter Pentium Dual-Core microprocessors, but with the first digit "1", instead of "2", suggesting their single-core functionality. A single-core Conroe-L with 1 MiB cache was deemed as not strong enough to distinguish the planned Pentiums from the Celerons, so it was replaced by dual-core CPUs, adding "Dual-Core" to the line's name. Throughout 2009, Intel changed the name back from Pentium Dual-Core to Pentium in its publications. Some processors were sold under both names, but the newer E5400 through E6800 desktop and SU4100/T4x00 mobile processors were not officially part of the Pentium Dual-Core line.

Comparison to the Pentium D

Although using the *Pentium* name, the desktop Pentium Dual-Core is based on the Core microarchitecture, which can clearly be seen when comparing the specification to the Pentium D, which is based on the NetBurst microarchitecture first introduced in the Pentium 4. Below the 2 or 4 MiB of shared-L2-cache-enabled Core 2 Duo, the desktop Pentium Dual-Core has 1 or 2 MiB of shared L2 Cache. In contrast, the Pentium D processors have either 2 or 4 MiB of non-shared L2 cache. Additionally, the fastest-clocked Pentium D has a factory boundary of 3.73 GHz, while the fastest-clocked desktop Pentium Dual-Core reaches 3.2 GHz. A major difference among these processors is that the desktop Pentium Dual Core processors have a TDP of only 65 W while the Pentium D ranges between 95 to 130 W. Despite the reduced clock speed, and lower amounts of cache, Pentium dual-core outperformed Pentium D by a fairly large margin.

Advantages

The proximity of multiple CPU cores on the same die allows the cache coherency circuitry to operate at a much higher clock-rate than is possible if the signals have to travel off-chip.

Combining equivalent CPUs on a single die significantly improves the performance of cache snoop (alternative: Bus snooping) operations. Put simply, this means that signals between different CPUs travel shorter distances, and therefore those signals degrade less. These higher-quality signals allow more data to be sent in a given time period, since individual signals can be shorter and do not need to be repeated as often. The largest boost in performance will likely be noticed in improved response-time while running CPU-intensive processes, like antivirus scans, ripping/burning media (requiring file conversion), or file searching. For example, if the automatic virus-scan runs while a movie is being watched, the application running the movie is far less likely to be starved of processor power, as the antivirus program will be assigned to a different processor core than the one running the movie playback.

Assuming that the die can fit into the package, physically, the multi-core CPU designs require much less printed circuit board (PCB) space than do multi-chip SMP designs. Also, a dual-core processor uses slightly less power than two coupled single-core processors, principally because of the decreased power required to drive signals external to the chip. Furthermore, the cores share some circuitry, like the L2 cache and the interface to the front side bus (FSB). In terms of competing technologies for the available silicon die area, multi-core design can make use of proven CPU core library designs and produce a product with lower risk of design error than devising a new wider core-design. Also, adding more cache suffers from diminishing returns.^[citation needed]

Multi-core chips also allow higher performance at lower energy. This can be a big factor in mobile devices that operate on batteries. Since each core in multi-core is generally more energy-efficient, the chip becomes more efficient than having a single large monolithic core. This allows higher performance with less energy. The challenge of writing parallel code clearly offsets this benefit.^[4]

Disadvantages

Maximizing the utilization of the computing resources provided by multi-core processors requires adjustments both to the operating system(OS) support and to existing application software. Also, the ability of multi-core processors to increase application performance depends on the use of multiple threads within applications. The situation is improving: for example the Valve Corporation's Source engine offers multi-core support,^{[5][6]} and Crytek has developed similar technologies for CryEngine 2, which powers their game, *Crysis*. Emergent Game Technologies' Gamebryo engine includes their Floodgate technology,^[7] which simplifies multicore development across game platforms. In addition, Apple Inc.'s second latest OS, Mac OS X Snow Leopard has a built-in multi-core facility called Grand Central Dispatch for Intel CPUs.

Integration of a multi-core chip drives chip production yields down and they are more difficult to manage thermally than lower-density single-chip designs. Intel has partially countered this first problem by creating its quad-core designs by combining two dual-core on a single die with a unified cache, hence any two working dual-core dies can be used, as opposed to producing four cores on a single die and requiring all four to work to produce a quad-core. From an architectural point of view, ultimately, single CPU designs may make better use of the silicon surface area than multiprocessing cores, so a development commitment to this architecture may carry the risk of obsolescence. Finally, raw processing

power is not the only constraint on system performance. Two processing cores sharing the same system bus and memory bandwidth limits the real-world performance advantage. If a single core is close to being memory-bandwidth limited, going to dual-core might only give 30% to 70% improvement. If memory bandwidth is not a problem, a 90% improvement can be expected^[citation needed]. It would be possible for an application that used two CPUs to end up running faster on one dual-core if communication between the CPUs was the limiting factor, which would count as more than 100% improvement.

Hardware

Trends

The general trend in processor development has moved from dual-, tri-, quad-, hexa-, octo-core chips to ones with tens or even hundreds of cores. In addition, multi-core chips mixed with simultaneous multithreading, memory-on-chip, and special-purpose "heterogeneous" cores promise further performance and efficiency gains, especially in processing multimedia, recognition and networking applications. There is also a trend of improving energy-efficiency by focusing on performance-per-watt with advanced fine-grain or ultra fine-grain power management and dynamic voltage and frequency scaling (i.e. laptop computers and portable media players).

Architecture

The composition and balance of the cores in multi-core architecture show great variety. Some architectures use one core design repeated consistently ("homogeneous"), while others use a mixture of different cores, each optimized for a different, "heterogeneous" role.

The article "CPU designers debate multi-core future" by Rick Merritt, EE Times 2008,^[8] includes these comments:

Chuck Moore suggested computers should be more like cellphones, using a variety of specialty cores to run modular software scheduled by a high-level applications programming interface.

Atsushi Hasegawa, a senior chief engineer at Renesas, generally agreed. He suggested the cellphone's use of many specialty cores working in concert is a good model for future multi-core designs.

Anant Agarwal, founder and chief executive of startup Tiler, took the opposing view. He said multi-core chips need to be homogeneous collections of general-purpose cores to keep the software model simple.

Managing concurrency acquires a central role in developing parallel applications. The basic steps in designing parallel applications are:

Partitioning

The partitioning stage of a design is intended to expose opportunities for parallel execution. Hence, the focus is on defining a large number of small tasks in order to yield what is termed a fine-grained decomposition of a problem.

Communication

The tasks generated by a partition are intended to execute concurrently but cannot, in general, execute independently. The computation to be performed in one task will

typically require data associated with another task. Data must then be transferred between tasks so as to allow computation to proceed. This information flow is specified in the communication phase of a design.

Agglomeration

In the third stage, development moves from the abstract toward the concrete. Developers revisit decisions made in the partitioning and communication phases with a view to obtaining an algorithm that will execute efficiently on some class of parallel computer. In particular, developers consider whether it is useful to combine, or agglomerate, tasks identified by the partitioning phase, so as to provide a smaller number of tasks, each of greater size. They also determine whether it is worthwhile to replicate data and/or computation.

Mapping

In the fourth and final stage of the design of parallel algorithms, the developers specify where each task is to execute. This mapping problem does not arise on uniprocessors or on shared-memory computers that provide automatic task scheduling.

On the other hand, on the server side, multicore processors are ideal because they allow many users to connect to a site simultaneously and have independent threads of execution. This allows for Web servers and application servers that have much better throughput.

PENTIUM PROCESSORS

Pentium is a brand used for a series of x86-compatible microprocessors produced by Intel. In its most current form, a Pentium processor is a consumer-level product with a two-star rating^[1], above the low-end Atom and Celeron products but below the faster Core i3, i5 and i7 lines as well as the high-end Xeon processors.

The name Pentium is originally derived from the Greek word *pente* (πέντε), meaning 'five' (as the series was Intel's 5th generation microarchitecture, the P5), and the Latin ending *-ium*. The current Pentium processors only share the name but are in fact based on the same processor chips that are used in the Intel Core but are typically used with a lower clock frequency, a partially disabled L3 cache and some of the advanced features such as Hyper-threading and Virtualization disabled.

Overview

During development Intel generally identifies processors with codenames, such as *Prescott*, *Willamette*, *Coppermine*, *Katmai*, *Klamath* or *Deschutes*. These usually become widely known,¹ even after the processors are given official names on launch.

History

The original Pentium branded CPUs were expected to be named 586 or i586, to follow the naming convention of previous generations (286, i386, i486). However, as the company wanted to prevent their competitors from branding their processors with similar names, as AMD had done with their Am486, Intel attempted to file a trademark on the name in the United States, only to be denied because a series of numbers was not considered distinct.

Following Intel's previous series of 8086, 80186, 80286, 80386, and 80486 microprocessors, the company's first P5-based processor was released as the original Intel Pentium on March 22, 1993. Due to its success, the Pentium brand would continue through several generations of high-end processors beyond the original. In 2006, the Pentium brand briefly disappeared from Intel's roadmaps,^{[3][4]} only to re-emerge in 2007.^[5]

In 1998, Intel introduced the Celeron^[6] brand for low-priced microprocessors. With the 2006 introduction of the Intel Core brand as the company's new flagship line of processors, the Pentium series was to be discontinued. However, due to a demand for mid-range dual-core processors, the Pentium brand was re-purposed to be Intel's mid-range processor series, in between the Celeron and Core series, continuing with the Pentium Dual-Core line.^{[7] [8][9]}

In 2009, the "Dual-Core" suffix was dropped, and new x86 microprocessors started carrying the plain *Pentium* name again.

Pentium-branded processors

P5 microarchitecture based

The original **Pentium** and **Pentium MMX** processors were the superscalar follow-on to the 80486 processor and were marketed from 1993 to 1999. Some versions of these were available as Pentium OverDrive that would fit into older CPU sockets.

Pentium

P6 microarchitecture based

In parallel with the P5 microarchitecture, Intel developed the **P6 microarchitecture** and started marketing it as the **Pentium Pro** for the high-end market in 1995. It introduced out-of-order execution and an integrated second level cache on dual-chip processor package. The second P6 generation replaced the original P5 with the **Pentium II** and rebranded the high-end version as **Pentium II Xeon**. It was followed by a third version called the **Pentium III** and **Pentium III Xeon**, respectively. The Pentium II line added the MMX instructions that were also present in the Pentium MMX.

Versions of these processors for the Laptop market were initially called **Mobile Pentium II** and **Mobile Pentium III**, later versions were called **Pentium III-M**. Starting with the Pentium II, the Celeron brand was used for low-end versions of most Pentium processors with a reduced feature set such as a smaller cache or missing power management features.

Pentium Pro

Pentium II

Pentium III

Netburst microarchitecture based

In 2000, Intel introduced a new microarchitecture called **NetBurst**, with a much longer pipeline enabling higher clock frequencies than the P6 based processors. Initially, these were called **Pentium 4** and the high-end versions have since been called just Xeon. As with Pentium III, there are both **Mobile Pentium 4** and **Pentium 4 M** processors for the laptop market, with Pentium 4 M denoting the more power-efficient versions. Enthusiasts version of the Pentium 4 with the highest clock frequency were called **Pentium 4 Extreme Edition**.

The **Pentium D** was the first multi-core Pentium, integrating two Pentium 4 chips in one package and was also available as the enthusiast **Pentium Extreme Edition**.

Pentium 4

Pentium D

[Pentium M microarchitecture based

In 2003, Intel introduced a new processor based on the P6 microarchitecture called **Pentium M**, which was much more power efficient than the Mobile Pentium 4, Pentium 4 M and Pentium III M. Dual-core version of the Pentium M was developed under the code name Yonah and sold under the marketing names Core Duo and **Pentium Dual-Core**. Unlike Pentium D, it integrated both cores on a single chip. From this point, the Intel Core brand name was used for the mainstream Intel processors and the Pentium brand became a low-end version between Celeron and Core. All Pentium M based designs including Yonah are for the mobile market.

Pentium M

Pentium Dual-Core

Core microarchitecture based

The **Pentium Dual-Core** name continued to be used when the Yonah design was extended with 64 bit support, now called the **Core microarchitecture**. This microarchitecture eventually replaced all NetBurst based processors across the four brands, Celeron, Pentium, Core and Xeon. Pentium Dual-Core processors based on the Core microarchitecture use the Allendale and Wolfdale-3M designs for desktop processors and Merom-2M for mobile processors.

Pentium Dual-Core

Pentium (2009)

In 2009, Intel changed the naming system for Pentium processors, renaming the Wolfdale-3M based processors to **Pentium**, without the Dual-Core name and introduced new single- and dual-core processors based on Penryn under the Pentium name.

The Penryn core is the successor to the Merom core and Intel's 45 nm version of their mobile series of Pentium microprocessors. The FSB is increased from 667 MHz to 800 MHz and the voltage is lowered. Intel released the first Penryn Core, the Pentium T4200, in December, 2008. In June 2009, Intel released the first single-core processor to use the Pentium name, a Consumer Ultra-Low Voltage (CULV) Penryn core called the Pentium SU2700.

In September 2009, Intel introduced the Pentium SU4000 series together with the Celeron SU2000 and Core 2 Duo SU7000 series, which are dual-core CULV processors based on Penryn-3M and using 800 MHz FSB. The Pentium SU4000 series has 2 MB L2 cache but is otherwise basically identical to the other two lines.

Nehalem microarchitecture based

The Nehalem microarchitecture was introduced in late 2008 as a successor to the Core microarchitecture, and in early 2010, a new **Pentium G6950** processor based on the **Clarkdale** design was introduced based on the **Westmere** refresh of Nehalem, which were followed by the mobile P6xxx based on Arrandale a few months later.

On January 7, 2010, Intel launched a new Pentium model using the Clarkdale chip in parallel with other desktop and mobile CPUs based on their new Westmere microarchitecture. The first model in this series is the Pentium G6950. The Clarkdale chip is also used in the Core i3-5xx and Core i5-6xx series and features a 32 nm process (as it is based on the Westmere

microarchitecture), integrated memory controller and 45 nm graphics controller and a third-level cache. In the Pentium series, some features of Clarkdale are disabled. Compared to Core i3, it lacks Hyper-Threading and the graphics controller in the Pentium runs at 533 MHz, while in the Core i3 i3-5xx series they run at 733 MHz. Dual Video Decode that enables Blu-ray picture-in picture hardware acceleration is disabled as well as Deep Color and xvYCCsupport¹. The memory controller in the Pentium supports DDR3-1066 max same as the Core i3 i3-5xx series (ref:<http://ark.intel.com/products/43529>). The L3 cache is also 1 MB less than in the Core i3-5xx series.

Sandy Bridge microarchitecture based

The Sandy Bridge microarchitecture was released in the Pentium line on May 22, 2011.

- ^aAll models share the following details: 2 cores, 2 logical processors (4 on Pentium 3xx with Hyper-threading), CPUID signature 206A7, family 6 (06h), model 42 (02Ah), stepping 7 (07h)
- ^bTLB / cache 64-byte Prefetching; Data TLB0 2-MB or 4-MB pages, 4-way associative, 32 entries; Data TLB 4-KB Pages, 4-way set associative, 64 entries; Instruction TLB 4-KB Pages, 4-way set associative, 128 entries, L2 TLB 1-MB, 4-way set associative, 64-byte line size; Shared 2nd-level TLB 4 KB pages, 4-way set associative, 512 entries.
- ^cAll models feature: On-chip Floating Point Unit, Enhanced Intel SpeedStep Technology (EIST), Intel 64, XD bit (an NX bitimplementation), Intel VT-x, Smart Cache.
- ^dAll models support: *MMX*, *SSE*, *SSE2*, *SSE3*, *SSSE3*, *SSE4.1*, *SSE4.2*
- ^eHD Graphics (Sandy Bridge) contain 6 EUs as well as HD Graphics 2000, but does not support the following technologies: Intel Quick Sync Video, InTru 3D, Clear Video HD, Wireless Display, and it doesn't support 3D Video or 3D graphics acceleration.

Pentium compatible Intel processors

Due to its prominence, the term "Pentium compatible" is often used to describe any x86 processor that supports the IA-32 instruction set and architecture. Even though they do not use the Pentium name, Intel also manufactures other processors based on the Pentium series for other markets. Most of these processors share the core design with one of the Pentium processor lines, usually differing in the amount of CPU cache, power efficiency or other features. The notable exception is the Atom line, which is an independent design.

- Celeron, a low-end version
- Core, the mainstream version including Core 2 and Core i7, now placed above Pentium
- Xeon, a high-end version used in servers and workstations
- A100 (discontinued), an ultra-mobile version of Pentium M
- EP80579, A system-on-a-chip based on Pentium M
- Atom, current ultra-mobile processors

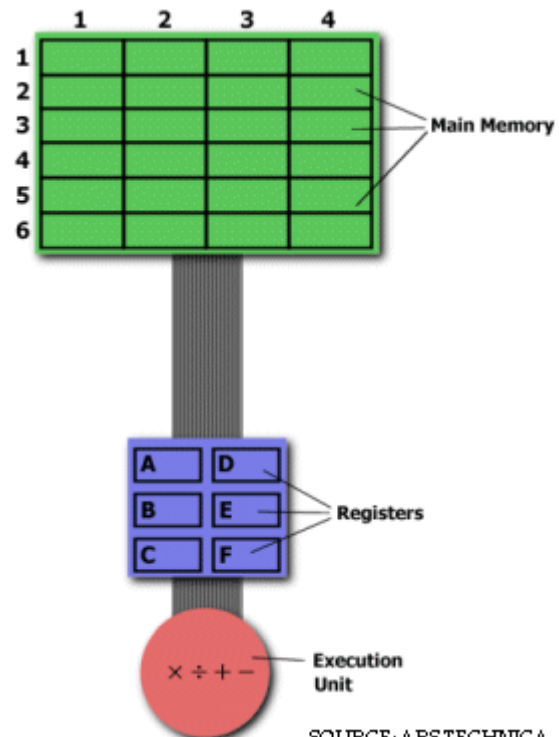
RISC ARCHITECTURE

risc vs. cisc

The simplest way to examine the advantages and disadvantages of RISC architecture is by contrasting it with its predecessor: CISC (Complex Instruction Set Computers) architecture.

Multiplying Two Numbers in Memory

On the right is a diagram representing the storage scheme for a generic computer. The main memory is divided into locations numbered from (row) 1: (column) 1 to (row) 6: (column) 4. The execution unit is responsible for carrying out all computations. However, the execution unit can only operate on data that has been loaded into one of the six registers (A, B, C, D, E, or F). Let's say we want to find the product of two numbers - one stored in location 2:3 and another stored in location 5:2 - and then store the product back in the location 2:3.



The CISC Approach

The primary goal of CISC architecture is to complete a task in as few lines of assembly as possible. This is achieved by building processor hardware that is capable of understanding and executing a series of operations. For this particular task, a CISC processor would come prepared with a specific instruction (we'll call it "MULT"). When executed, this instruction loads the two values into separate registers, multiplies the operands in the execution unit, and then stores the product in the appropriate register. Thus, the entire task of multiplying two numbers can be completed with one instruction:

MULT 2:3, 5:2

MULT is what is known as a "complex instruction." It operates directly on the computer's memory banks and does not require the programmer to explicitly call any loading or storing functions. It closely resembles a command in a higher level language. For instance, if we let "a" represent the value of 2:3 and "b" represent the value of 5:2, then this command is identical to the C statement "a = a * b."

One of the primary advantages of this system is that the compiler has to do very little work to translate a high-level language statement into assembly. Because the length of the code is relatively short, very little RAM is required to store instructions. The emphasis is put on building complex instructions directly into the hardware.

The RISC Approach

RISC processors only use simple instructions that can be executed within one clock cycle. Thus, the "MULT" command described above could be divided into three separate

commands: "LOAD," which moves data from the memory bank to a register, "PROD," which finds the product of two operands located within the registers, and "STORE," which moves data from a register to the memory banks. In order to perform the exact series of steps described in the CISC approach, a programmer would need to code four lines of assembly:

```
LOAD A, 2:3
LOAD B, 5:2
PROD A, B
STORE 2:3, A
```

At first, this may seem like a much less efficient way of completing the operation. Because there are more lines of code, more RAM is needed to store the assembly level instructions. The compiler must also perform more work to convert a high-level language statement into code of this form.

CISC	RISC	
Emphasis on hardware	Emphasis on software	However, the RISC strategy also brings some very important advantages. Because each instruction requires only one clock cycle to execute, the entire program will execute in approximately the same amount of time as the multi-cycle "MULT" command. These RISC "reduced instructions" require less transistors of hardware space than the complex instructions, leaving more room for general purpose registers. Because all of the instructions execute in a uniform amount of time (i.e. one clock), pipelining is possible.
Includes multi-clock complex instructions	Single-clock, reduced instruction only	
Memory-to-memory: "LOAD" and "STORE" incorporated in instructions	Register to register: "LOAD" and "STORE" are independent instructions	
Small code sizes, high cycles per second	Low cycles per second, large code sizes	
Transistors used for storing complex instructions	Spends more transistors on memory registers	

Separating the "LOAD" and "STORE" instructions actually reduces the amount of work that the computer must perform. After a CISC-style "MULT" command is executed, the processor automatically erases the registers. If one of the operands needs to be used for another computation, the processor must re-load the data from the memory bank into a register. In RISC, the operand will remain in the register until another value is loaded in its place.

The Performance Equation

The following equation is commonly used for expressing a computer's performance ability:

$$\frac{\text{time}}{\text{program}} = \frac{\text{time}}{\text{cycle}} \times \frac{\text{cycles}}{\text{instruction}} \times \frac{\text{instructions}}{\text{program}}$$

The CISC approach attempts to minimize the number of instructions per program,

sacrificing the number of cycles per instruction. RISC does the opposite, reducing the cycles per instruction at the cost of the number of instructions per program.

RISC Roadblocks

Despite the advantages of RISC based processing, RISC chips took over a decade to gain a foothold in the commercial world. This was largely due to a lack of software support.

Although Apple's Power Macintosh line featured RISC-based chips and Windows NT was RISC compatible, Windows 3.1 and Windows 95 were designed with CISC processors in mind. Many companies were unwilling to take a chance with the emerging RISC technology. Without commercial interest, processor developers were unable to manufacture RISC chips in large enough volumes to make their price competitive.



Another major setback was the presence of Intel. Although their CISC chips were becoming increasingly unwieldy and difficult to develop, Intel had the resources to plow through development and produce powerful processors. Although RISC chips might surpass Intel's efforts in specific areas, the differences were not great enough to persuade buyers to change technologies.

The Overall RISC Advantage

Today, the Intel x86 is arguable the only chip which retains CISC architecture. This is primarily due to advancements in other areas of computer technology. The price of RAM has decreased dramatically. In 1977, 1MB of DRAM cost about \$5,000. By 1994, the same amount of memory cost only \$6 (when adjusted for inflation). Compiler technology has also become more sophisticated, so that the RISC use of RAM and emphasis on software has become ideal.

[what is risc?](#) · [mips](#) · [pipelining](#) · [recent developments](#) · [bibliography](#) · [about this site](#)

Atmel AVR RISC microcontroller:

The AVR is a [modified Harvard architecture 8-bit RISC](#) single chip [microcontroller](#) which was developed by [Atmel](#) in 1996. The AVR was one of the first microcontroller families to use on-chip [flash memory](#) for program storage, as opposed to [one-time programmable ROM](#), [EPROM](#), or [EEPROM](#) used by other microcontrollers at the time. However, it is commonly accepted that AVR stands for Alf (Egil Bogen) and Vegard (Wollan)'s Risc processor" the use of "AVR" generally refers to the 8-bit RISC line of Atmel AVR Microcontrollers.

Among the first of the AVR line was the AT90S8515, which in a 40-pin DIP package has the same pinout as an [8051](#) microcontroller, including the external multiplexed address and data bus. The polarity of the RESET line was opposite (8051's having an active-high RESET, while the AVR has an active-low RESET) but other than that, the pinout was identical.

overview

The AVR is a modified Harvard architecture machine where program and data are stored in separate physical memory systems that appear in different address spaces, but having the ability to read data items from program memory using special instructions.

Basic families

AVRs are generally classified into five broad groups:

- **tinyAVR** — the ATtiny series
 - 0.5–8 kB program memory
 - 6–32-pin package
 - Limited peripheral set
- **megaAVR** — the ATmega series
 - 4–256 kB program memory
 - 28–100-pin package
 - Extended instruction set (Multiply instructions and instructions for handling larger program memories)
 - Extensive peripheral set
- **XMEGA** — the ATxmega series
 - 16–384 kB program memory
 - 44–64–100-pin package (A4, A3, A1)
 - Extended performance features, such as DMA, "Event System", and cryptography support.
 - Extensive peripheral set with DACs
- **Application-specific AVR**
 - megaAVRs with special features not found on the other members of the AVR family, such as LCD controller, [USB](#) controller, advanced PWM, CAN etc.
- **FPSLIC™ (AVR with FPGA)**
 - [FPGA](#) 5K to 40K gates
 - SRAM for the AVR program code, unlike all other AVRs
 - AVR core can run at up to 50 MHz ^[5]

- **32-bit AVR**s

In 2006 Atmel released microcontrollers based on the new, 32-bit, [AVR32](#) architecture. They include [SIMD](#) and [DSP](#) instructions, along with other audio and video processing features. This 32-bit family of devices is intended to compete with the [ARM](#) based processors. The instruction set is similar to other RISC cores, but is not compatible with the original AVR or any of the various ARM cores.

Architecture

[Flash](#), [EEPROM](#), and [SRAM](#) are all integrated onto a single chip, removing the need for external memory in most applications. Some devices have a parallel external bus option to allow adding additional data memory or memory-mapped devices. Almost all devices (except the smallest TinyAVR chips) have serial interfaces, which can be used to connect larger serial EEPROMs or flash chips. The fast-access register file concept contains 32 x 8-bit general-purpose working registers With a single clock cycle access time. This means that during one single clock cycle, one Arithmetic Logic Unit (ALU) operation is executed. Two operands are output from the register file, the operation is executed and the result is stored back in the register file—in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect addresses register pointers for Data Space addressing, enabling efficient address calculations. One of the three address pointers is also used as the address pointer for the constant table look-up function.

These added function registers are the 16-bit X-register, Y-register, and Z-register.

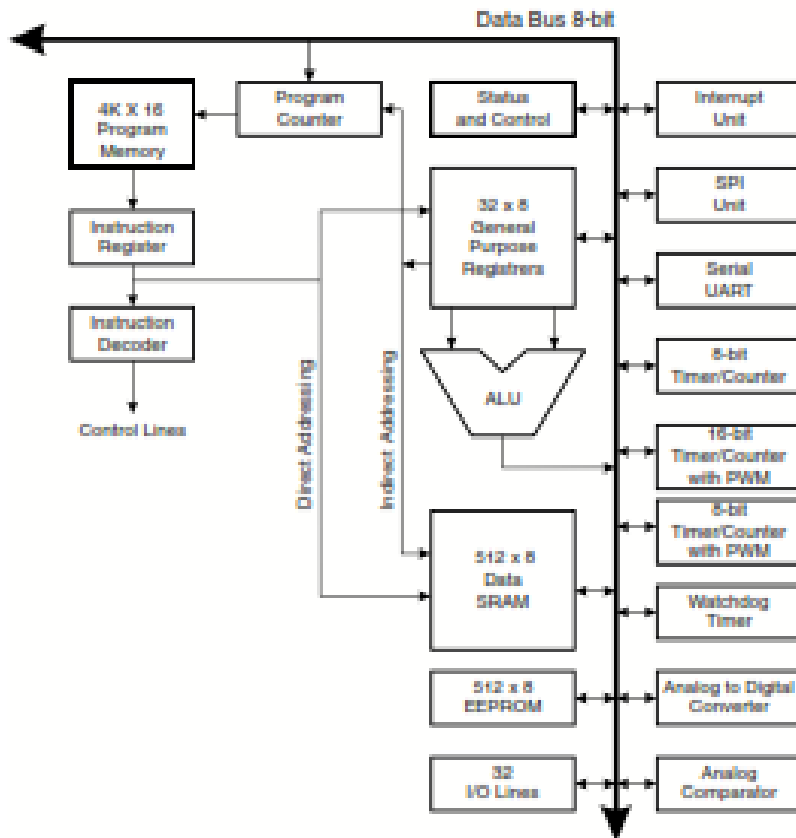


Figure. The AT90S8535 AVR RISC Architecture

The ALU supports arithmetic and logic functions between registers or between a constant

And a register. Single register operations are also executed in the ALU. Figure 4 shows the AT90S8535 AVR RISC microcontroller architecture.

PIC MICROCONTROLLER

INTRODUCTION:

PIC microcontrollers are popular processors developed by Microchip Technology with built-in RAM, memory, internal bus, and peripherals that can be used for many applications. PIC originally stood for “Programmable Intelligent Computer” but is now generally regarded as a “Peripheral Interface Controller”.

PIC microcontrollers can be programmed in Assembly, C or a combination of the two. Other high-level programming languages can be used but embedded systems software is primarily written in C.

TYPES OF PIC’S:

PIC microcontrollers are broken up into two major categories: 8-bit microcontrollers and 16-bit microcontrollers. Each category is further subdivided into product families as shown in the following table:

8-bit MCU Product Family	16-bit MCU Product Family
PIC10 PIC12 PIC14 PIC16 PIC18	PIC24F PIC24H dsPIC30 dsPIC33

The microcontrollers in the PIC10 through PIC14 families are considered low-end microcontrollers. PIC microcontrollers in the PIC16 and PIC18 families are considered mid-level microcontrollers while 16-bit PICs are considered high-end microcontrollers.

FEATURES:

- Family includes controllers from 16c61/62/64/71/74/710/715 etc.
- They are RISC processors and uses Harvard architecture.
- Different bus widths of data and program memory.
- Data memory is 8 bit wide where as program memory is 12,14,16 bits wide.
- The instruction holds immediate data along with instruction code.
- Only 35 instructions.
- Most instructions take 0.2 microseconds to execute when operated at 20 MHz.
- Machine cycle consist of 4 clock pulses.
- Instruction set is highly orthogonal.
- 1-3 Timers with 8/16 bit prescaler.
- Watch Dog timer (WDT)
- 13-33 I/O pins.
- 3-12 interrupt Sources.
- 4/8 Channel, 8 bit on chip ADC.
- Power on Reset. (POR)
- Brown out Reset (BOR).
- Capture/Compare/ PWM modules.
- USART
- Synchronous serial port (SSP) with SPI and I2C.
- Power saving SLEEP mode.
- Wide operating Voltage range 2.5 V to 6.0 V. Very Low power consumption.
- Commercial, Industrial and Extended Temperature ranges.
- Parallel slave port (PSP), 8 bits wide with external RD, WR and CS controls.

PICs also come in several types of packages:

Plastic Dual Inline Package (PDIP), Small-Outline Transistor (SOT), Dual Flat No-lead (DFN), Mini Small Outline Package (MSOP), Thin Quad Flat Pack (TQFP), Plastic Leaded Chip Carrier (PLCC), CERamic QUADpack (CERQUAD),

The reason for the number of packages is that there are some PICs with 100 I/O pins! The microcontrollers are basically rectangular or square shaped. The easiest package to work with is DIP or PDIP because it is easily bread boardable and can easily be soldered. Writing your code completely in C because it is much faster and easier than writing your code in Assembly or a combination of languages.

PIC ARCHITECTURE:

ALU

Size is 8 bit Performs operations with temporary working register and (W register) and any register file. W register 8 bit wide. It contains one of source operands during execution of instruction and may serve as the destination for the result of operation. Used only for ALU operations.

STATUS Register

7	6	5	4	3	2	1	0
0	0	RP0	TO	PD	Z	DC	C

C = Carry Bit

DC = Digits Carry (Same as AC)

Z = Zero

TO = Reset status bit (Time out bit)

PD = Reset status bit (Power down)

These bits are used along with SLEEP mode. After coming out from SLEEP processor checks these bits to determine which kind of event is responsible for bringing out of SLEEP mode. RP0 = Register Bank Select bit. If 0 selects bank 0 otherwise bank 1.

FSR Register (File Selection Register):

FSR is a pointer used for indirect memory addressing in the whole register file. In indirect addressing mode one has to write address byte in FSR and then use INDF (Indirect thro FSR). INDF is used in instruction.

PCLATH (PC Latch):

Can be independently read or written like any other register. It's different from PC and is separate entity. It is 5 bits. This is added with PCL (Program Counter Lower) so as to get 13 bit address.

PROGRAM MEMORY

- PIC 16c6X/7X is 2K or 4K.
- 11 bit or 12 bit address is used out of
- 13 bits in PC.
- Maximum memory that can be accessed is 8K.
- After reset program counter is cleared.
- At 0000h there is "goto Mainline" Instruction which takes PC to 0005h.

DATA MEMORY:

Register File Structure: They are the memory locations that are addressed by instruction. There is general purpose and special purpose register file. General purpose are 8 bit RAM locations and special purpose are I/O ports and control registers.

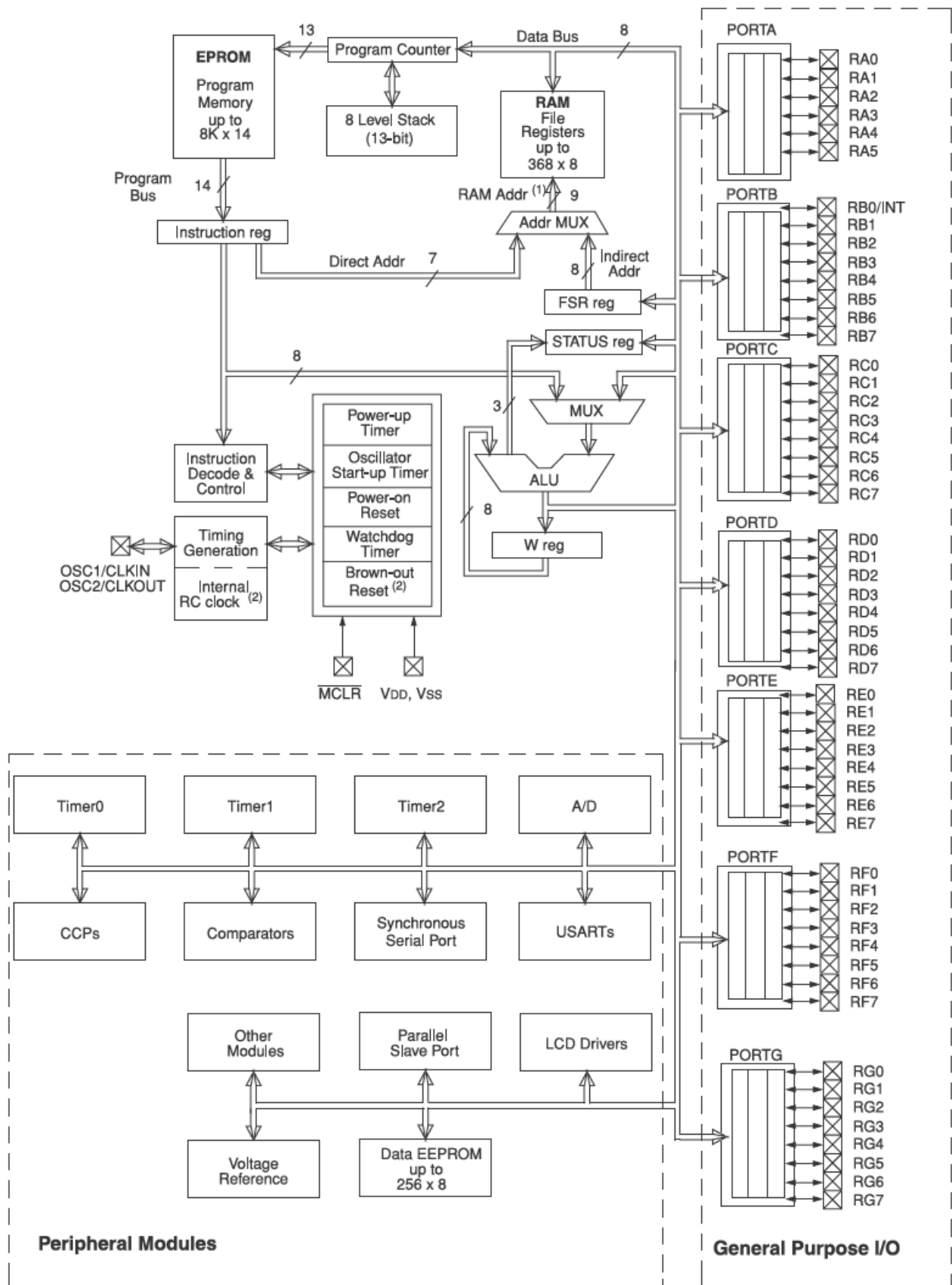
ADDRESSING MODES:

Direct Addressing:

It uses 7 bits of instruction and the 8 bit from RP0. If bit is 0 then bank 0 otherwise bank 1.

Indirect addressing

In this mode the 8 bit address of the location in register file to be accessed is written in FSR and use INDF.



16. University previous Question papers

Code No: 07A6EC02

R07

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

B. Tech III Year II Semester Examinations, May - 2013

Microprocessors and Microcontrollers
(Electrical and Electronics Engineering)

Time: 3 hours

Max. Marks: 80

Answer any five questions
All questions carry equal marks

1. What is meant by an addressing mode? Explain the different addressing modes supported by 8086 with suitable examples. [16]
- 2.a) Write an 8086 assembly language program to arrange a given series of hexadecimal bytes in ascending order.
- b) Explain the following assembler directives:
i) SEGMENT, ii) ORG,
iii) OFFSET, iv) DW [8+8]
- 3.a) Explain the control word format of 8255 in I/O and BSR mode.
- b) Explain the A/D converter interface to 8086 microprocessor. [8+8]
- 4.a) With the help of the internal block diagram, explain the working of 8259 priority interrupt controller.
- b) Explain the various hardware and software interrupts in 8086 microprocessor. [8+8]
- 5.a) Give an overview of RS-232C serial data standard.
- b) Write short notes on IEEE-488 standard. [8+8]
6. Draw and explain the internal architecture of 8051 microcontroller. [16]
- 7.a) Explain the format and bit definitions of the following SFRs in 8051:
i) TMOD, ii) SCON
- b) Explain the interrupt structure of 8051. [8+8]
8. List out the various human and keyboard factors that effect the keyboard application programs. Discuss how a lead per key keyboard is interfaced to the 8051 microcontroller with the help of a diagram and explain the "Getkey" routine for scanning. [16]

Code No: 09A60204

R09

SET-1

B. Tech III Year II Semester Examinations, April/May - 2012

MICROPROCESSORS AND MICROCONTROLLERS

(Common to EEE, ECE, EIE, ETM, ECM, ICE)

Time: 3 hours

Max. Marks: 75

Answer any five questions

All questions carry equal marks

1. Explain different registers used in 8086 and its memory segmentation. What are the registers used to access memory. [15]
2. Explain different addressing modes in 8086. [15]
3. Show 8255 PPI in mode1 operation and interface to 8086. [15]
4. Explain the interrupt structure of 8086. [15]
5. Explain the need for RS232C interface. Explain serial communication standards with respect to voltage levels. [15]
6. What is the difference between microprocessor and microcontroller? Give 8051 architecture. [15]
7. Explain how the interrupts are used in real-time. [15]
8. Explain standard AVR architecture. [15]

--ooOoo--

Code No: 09A60204

R09

SET-2

B. Tech III Year II Semester Examinations, April/May - 2012

MICROPROCESSORS AND MICROCONTROLLERS

(Common to EEE, ECE, EIE, ETM, ECM, ICE)

Time: 3 hours

Max. Marks: 75

Answer any five questions

All questions carry equal marks

1. Explain memory mapping techniques in 8086. [15]
2. Explain instruction format for 8086. [15]
3. Interface 8086 to keyboard and display unit. [15]
4. Explain interrupt service routine concept. [15]
5. Explain 8251 USART. [15]
6. Explain the timers in 8051 and its modes. [15]
7. Explain the timers and counters in 8051. [15]
8. Explain memory organization in AVR. [15]

--00000--

Code No: 09A60204

R09

SET-3

B. Tech III Year II Semester Examinations, April/May - 2012

MICROPROCESSORS AND MICROCONTROLLERS

(Common to EEE, ECE, EIE, ETM, ECM, ICE)

Time: 3 hours

Max. Marks: 75

Answer any five questions

All questions carry equal marks

1. Explain interrupts of 8086 its minimum and maximum mode signals and common function signals. [15]
2. What are assembler directives and macros? Consider one example and show how they are used? [15]
3. Explain A/D and D/A conversion mechanism. [15]
4. Explain 8259 interrupt controller along with control registers. [15]
5. Explain the interfacing of RS-232 to 8086. [15]
6. Explain internal and external memory of 8051. [15]
7. Explain programming serial communication interrupts. [15]
8. Explain interrupt structure in AVR. [15]

--00000--

Code No: 09A60204

R09

SET-4

B. Tech III Year II Semester Examinations, April/May - 2012

MICROPROCESSORS AND MICROCONTROLLERS

(Common to EEE, ECE, EIE, ETM, ECM, ICE)

Time: 3 hours

Max. Marks: 75

Answer any five questions
All questions carry equal marks

1. Explain the architecture of 8086. [15]
2. Explain sorting technique with an example. [15]
3. Explain I/O mapped I/O and memory mapped I/O. [15]
4. Explain the interface of 8257 to 8086. [15]
5. Explain IEEE-488 bus and its protocols. [15]
6. Give any five instructions of 8051 and explain each. [15]
7. Explain special function registers in 8051 and show how they are used for programming. [15]
8. Explain register file of AVR. [15]

--00000--

R09

Code No: 09A60204

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

B. Tech III Year II Semester Examinations, May – 2013

Microprocessors and Microcontrollers

(Common to EEE, ECE, EIE, ETM, E.COMP.E, ICE)

Time: 3 hours

Max. Marks: 75

Answer any five questions

All questions carry equal marks

1.a) With a neat block diagram explain the architecture of 8086 processor.

b) Explain the following instructions of 8086

i) TEST

ii) CMP

iii) XLAT

iv) CBW.

[8+7]

2.a) What is the purpose of the Instruction Queue of BIU in 8086? Why is the size of the queue limited to 6 Bytes?

b) Explain addressing modes of 8086 with an example.

[5+10]

3. An 8086-1 system with 8255 is interfaced at port A address FFF8H and Port C address at FFFE H. 8086-2 system with another 8255 is interfaced at port A address FFF8H and Port C address at FFFE H. Give necessary hardware and software for transferring 10 bytes of data in parallel from 8086-1 to another 8086-2. Assume that both systems run on the same clock rate.

[15]

4.a) Explain how to interface Digital-to-Analog Converter (DAC) to 8086 processor. Give the hardware and software for it.

b) Explain type 3 interrupt in 8086.

[12+3]

5.a) Draw and explain interrupt acknowledgement cycle of 8086.

b) Explain the mode word format and the command word format of 8251A. [5+10]

6.a) Assume that ROM space of 8051 starting at 250H contains "Hello", write a program to transfer the bytes into RAM locations starting at 40H.

b) Explain the stack operation in 8051 microcontroller and also discuss necessary instructions to access the stack.

[9+6]

7.a) List out the steps involved in programming the 8051 to transfer data serially.

b) Write an 8051 program to find Y where $Y = x^2 + 5$ and x is between 0 and 5.

[5+10]

8.a) Explain the salient features of AVR RISC controller with reference to its architecture.

b) Discuss the interrupt structure of RISC controller.

[15]

III B. Tech II Semester Regular Examinations, Apr/May 2010

MICROPROCESSORS AND INTERFACING (Common to ECE, BME, EIE)

Time: 3 Hours

Max Marks: 80

Answer Any FIV Questions

All Questions Carry Equal Marks

1. a) Draw and explain the flag register of 8086 microprocessor.
b) Explain the different logical instructions of 8086 microprocessor. [8+8]
2. Write an assembly language program in 8086 to arrange the given series of hexadecimal numbers in ascending order. [16]
3. With neat circuit diagram and timing diagrams, explain the minimum mode operation of 8086 microprocessor. [16]
4. Interface an 8255 with 8086 to work as an I/O port. Initialize port A as output port, port B as input port and port C as output port. Port A address should be 0740H. Write a program to sense switch positions $SW_0 - SW_7$ connected at port B. The sensed pattern is to be displayed in port A, to which 8 LEDs are connected, which the port C lower displays number of on switches out of the total eight switches. [16]
5. a) What is the difference between maskable and non-maskable interrupts?
Give some examples?
b) Discuss about the following control word formats of 8259:
 - i. Initialization Command Words (ICWs).
 - ii. Operational Command Words (OCWs) [6+10]
6. a) Discuss the types of serial communication.
b) With a neat circuit diagram, explain the interfacing of 8251 with 8086. [6+10]
7. a) What is meant by paging? Explain its advantages and disadvantages.
b) Explain the procedure of converting linear address into physical address. [8+8]

8. a) Discuss the advantages of microcontroller based systems over microprocessor based systems.

b) Draw and discuss the formats and bit definitions of the following SFRs

i. PCON ii. TCON iii. TMOD [4+12]

III B. Tech II Semester Regular Examinations, Apr/May 2010

MICROPROCESSORS AND INTERFACING (Common to ECE, BME, EIE)

Time: 3 Hours

Max Marks: 80

Answer Any FIVE Questions

All Questions Carry Equal Marks

1. Explain different addressing modes of 8086 microprocessors with suitable examples. [16]
2. a) Write an assembly language program in 8086 for the addition of a series of 8-bit numbers.
b) Write an assembly language program to display a message "J N T U KAKINADA" on the CRT stream of a micro computer. [8+8]
3. Draw the functional pin diagram of 8086 and explain the functions of different pins. [16]
4. Interface a 4*4 keyboard with 8086 using 8255 and write an ALP for detecting a key closure and return the key code in AL. The debouncing period for a key is 10ms. Use software key bouncing technique. [16]
5. What is interrupt service routine? Draw the interrupt vector table and explain different interrupts available in 8086. [16]
6. a) Discuss the asynchronous and synchronous data transfer schemes. b) Explain the interfacing of 8251 with 8086 with necessary circuit diagram. [8+8]

- a) Explain the salient features of the 80386 microprocessors.
 - b) Explain how the linear address is converted into physical address[8+8]
8. a) Discuss the register set of 8051 microcontroller.
- b) Explain the interrupt structure of the 8051 microcontroller.[8+8]

III B. Tech II Semester Regular Examinations, Apr/May 2010

MICROPROCESSORS AND INTERFACING (Common to ECE, BME, EIE)

Time: 3 Hours

Max Marks: 80

Answer Any FIVE Questions

All Questions Carry Equal Marks

1. List the register set of 8086 microprocessor and explain the functions of each of them.[16]
2. a) Write an ALP in 8086 for the addition of a series of 8-bit numbers.
b) Write an ALP in 8086 to move a string of data words from offset 2000H to offset 3000H and the length of the string in 0FH. [8+8]
3. With neat circuit and timing diagrams, explain the maximum mode operation of 8086 microprocessor. [16]
4. Interface ADC 0808 with 8086 using 8255 ports. Use port A of 8255 for transferring digital data output of ADC to the CPU and port C for control signals. Assume that an analog input is present at I/P2 of the ADC and a clock input of suitable frequency is available for ADC. Draw the schematic and write required ALP. [16]
5. Draw and explain the architecture of 8259 programmable Interrupt Controller. [16]
6. a) Explain different modes data transmission.
b) Draw and explain the internal architecture of 8251 USART[8+8]
7. a) Differentiate RISC and CISC processors

b) Explain the real mode and protected mode operations of 80386 microprocessor. [6+10]

8. a) Discuss various addressing modes of 8051.

b) Explain in detail about serial port operation in 8051 microcontroller. [8+8]

III B. Tech II Semester Regular Examinations, Apr/May 2010

MICROPROCESSORS AND INTERFACING (Common to ECE, BME, EIE)

Time: 3 Hours

Max Marks: 80

Answer Any FIVE Questions
All Questions Carry Equal Marks

1. a) Explain different data transfer instructions of 8086 microprocessor. b) Define and explain macros. [10+6]
2. a) Write an assembly language program in 8086 to find out the largest numbers from a given unordered array of 8-bit numbers, stored in the locations starting from a known address.
b) Write an assembly language program in 8086 to perform a one bit BCD addition. [8+8]
3. a) What is DMA? Explain the need for DMA.
b) With a neat block diagram, explain the working of 8257 DMA controller. [8+8]
4. Interface DAC AD7532 with an 8086 CPU running at 8MHz and write an assembly language program to generate a saw tooth waveform of period 1ms with V_{max} 5V. [16]
5. a) Describe some important features of 8259 interrupt controller.
b) Distinguish between Master and Slave mode operation of 8259. [8+8]
6. Design a hardware interfacing circuit for interfacing 8251 with 8086. Set the 8251A in asynchronous mode as a transmitter and receiver with even

parity enabled, 2 stop bits, 8-bit character length, frequency 160 KHz and baud rate 10K. Write an ALP to transmit 100 bytes of data string starting at location 2000:5000H. [16]

7. a) Explain the salient features of the Pentium processor.

b) Differentiate the paging and segmentation.[8+8]

8. Draw and discuss the formats and bit definitions of the following SFRs

i. PCON

ii. TCON

iii. TMOD

iv. SCON [4+4+4+4]

III B. Tech II Semester Regular Examinations, NOV/DEC 2009

MICROPROCESSORS AND INTERFACING

(Common to Electronics & Communication Engineering,
Electronics & Instrumentation Engineering, Bio-Medical
Engineering, Electronics & Control Engineering and Electronics &
Telematics)

Time: 3 hours

Max Marks: 80

Answer any FIVE Questions

All Questions carry equal marks

1. (a) Draw the architectural diagram of 8085 and explain the function of each block in detail

(b) Discuss about Multiplexing in 8086 microprocessor [10+6]

2. (a) Discuss about the following addressing modes of 8086 with some examples i. Directing addressing

ii. Register addressing

iii. Immediate addressing

iv. Based addressing with displacement

- (b) Explain why assembler directives are required? [8+8]
3. (a) Write a program in 8086 to add two 8-bytes of data available in memory location array1 and array 2. Store the result in array3
- (b) Write an ALP to count number of 0s in a 16 bit binary string [10+6]
4. (a) With a neat pin diagram explain the minimum mode operation of 8086 (08)
- (b) With a neat timing diagram explain how a READ operation is performed by 8086[8+8]
5. (a) Explain the following pins of 8255
- i. A0 and A1
 - ii. WR
 - iii. RESET
 - iv. PC0-PC7
- (b) Explain the use of handshaking signals used in 8255 [8+8]
6. (a) How many Initialization Command words are required for a single 8259
- In an 8086 based system? Explain their format?
- (b) Discuss the following interrupts?
- i. Single step Execution
 - ii. Interrupt on Overflow. [10+6]
7. (a) Draw the circuit of TTL to RS232 conversion and explain the necessity of this interface.
- (b) Draw the necessary circuit to interface 8251 to an 8086 based system with an address A0H. Write the sequence of instructions to initialize 8251 for syn-chronous transmission with odd parity, single SYNC character, 8bit data character.[6+10]
8. (a) Explain the internal and external program memory as well as data memory of 8051 with the diagram showing their capacities.
- (b) Draw the diagram to Interface Program memory of 16K x 8 EPROM to 8051 and give its memory map. The address of memory map should start from 0000H.[8+8]

III B.Tech II Semester Supplementary Examinations, Nov/Dec 2009

MICROPROCESSORS AND INTERFACING

(Common to Electronics & Communication Engineering,
Electronics & Instrumentation Engineering, Bio-Medical
Engineering, Electronics & Control Engineering and Electronics &
Telematics)

Time: 3 hours

Max Marks: 80

Answer any FIVE Questions

All Questions carry equal marks

1. (a) With a neat architectural diagram explain the functioning of an 8086 micro-processor
(b) Compare the flag registers of 8086 & 8085 [10+6]
2. (a) Explain the following 8086 instructions with examples. i. MUL
ii. IMUL
iii. DIV
iv. IDIV
(b) Differentiate between procedures and macros using certain examples. [8+8]
3. (a) Write a program in 8086 to add two 8-bytes of data available in memory location array1 and array 2. Store the result in array3
(b) Write an ALP to count number of 0s in a 16 bit binary string [10+6]
4. (a) Explain how static RAMs are interfaced to 8086. Give necessary interface diagram assuming appropriate signals and memory size
(b) Explain the need of DMA. Discuss in detail about DMA data transfer method [8+8]
5. (a) Explain the following pins of 8255. i. A0 and A1
ii. WR

iii. RESET

iv. PC0-PC7

(b) Explain the use of handshaking signals used in 8255 [8+8]

6. (a) How many Initialization Command words are required for a single 8259 in an 8086 based system? Explain their format?

(b) Discuss the following interrupts?

i. Single step Execution

ii. Interrupt on Overflow.

[10+6]

7. (a) Explain why serial data transfer is mostly preferred over parallel data transfer. Give reasons.
- (b) Distinguish between data formats used for Synchronous and Asynchronous serial data transfer modes. [8+8]
8. (a) Draw and discuss the formats and bit definitions of the following registers of 8051.
 - i. IP
 - ii. IE
- (b) The internal RAM memory (128 bytes) is divided into three parts. Explain it with neat diagram? [8+8]

III B.Tech II Semester Supplementary Examinations, Nov/Dec 2009

MICROPROCESSORS AND INTERFACING

(Common to Electronics & Communication Engineering,
Electronics & Instrumentation Engineering, Bio-Medical
Engineering, Electronics & Control Engineering and Electronics &
Telematics)

Time: 3 hours

Max Marks: 80

Answer any FIVE Questions

All Questions carry equal marks

1. (a) Draw the architectural diagram of 8085 and explain the function of each block in detail
- (b) Discuss about Multiplexing in 8086 microprocessor [10+6]
2. (a) Discuss about the following addressing modes of 8086 with some examples
 - i. Directing addressing
 - ii. Register addressing
 - iii. Immediate addressing
 - iv. Based addressing with displacement
- (b) Explain why assembler directives are required? [8+8]
3. (a) Write an ALP in 8086 to sort a given set of 8-bit unsigned integers into as- cending order by bubble sort method
- (b) Write an ALP in 8086 to display the string "WELCOME" on the screen [10+6]

4. (a) With relevant pin diagrams explain the minimum and maximum mode operations of 8086
(b) Explain briefly about DMA data transfer method. [12+4]
5. (a) Explain the modes of operation of 8255 with relevant diagrams?
(b) With a neat diagram explain how a 7-segment display is interfaced to 8086 using 8255? [8+8]
6. (a) Differentiate between Initialization Command Words and Operation Command Words of 8259.
(b) Discuss about the interrupt priority schemes used in 8259. [10+6]
7. (a) Draw the circuit of RS232 to TTL conversion and explain this interface?
(b) Draw the internal block diagram of 8251 USART and explain in detail about each block. [6+10]
8. (a) Explain the alternate functions of Port 0, Port 2 and Port 3.
(b) Draw and discuss the formats and bit definitions of TCON register in 8051. [8+8]

III B.Tech II Semester Supplementary Examinations, Nov/Dec 2009

MICROPROCESSORS AND INTERFACING

(Common to Electronics & Communication Engineering,
Electronics & Instrumentation Engineering, Bio-Medical
Engineering, Electronics & Control Engineering and Electronics &
Telematics)

Time: 3 hours

Max Marks: 80

Answer any FIVE Questions

All Questions carry equal
marks

1. Explain the function of following registers in 8086 microprocessor :

[4×4=16] (a)

AX,BX,CX,DX

(b) CS,DS,SS,ES

- (c) BP,SP,SI,DI
- (d) IP and Instruction Queue
2. (a) Discuss about the following addressing modes of 8086 with some examples
- Directing addressing
 - Register addressing
 - Immediate addressing
 - Based addressing with displacement
- (b) Explain why assembler directives are required? [8+8]
3. (a) Write a program in 8086 to add two 8-bytes of data available in memory location array1 and array 2. Store the result in array3
- (b) Write an ALP to count number of 0s in a 16 bit binary string [10+6]
4. (a) With relevant pin diagrams explain the minimum and maximum mode operations of 8086
- (b) Explain briefly about DMA data transfer method. [12+4]
5. (a) Write an ALP in 8086 to generate triangular waveform and give the necessary circuit setup with a DAC.
- (b) Write an algorithm for driving a stepper motor. Assume that the desired direction stored in BL and the number of steps is stored in CL. Write a delay routine for 1 millisecond after each step movement. [8+8]
6. (a) How many Initialization Command words are required for a single 8259 in an 8086 based system? Explain their format? (b) Discuss the following interrupts?
- Single step Execution
 - Interrupt on Overflow. [10+6]
7. (a) Explain with a neat diagram the working of 8251 PCI. (b) Draw the interface circuits for data conversion from

- i. TTL to RS232C and
 - ii. RS232C to TTL [8+8]
8. (a) Discuss about various addressing modes of 8051.
- (b) Explain the interrupt structure of 8051 [8+8]

III B.Tech I Semester Examinations, November 2010

MICROPROCESSORS AND INTERFACING

Common to Information Technology, Instrumentation And
Control Engineering, Electronics And Computer Engineering,
Computer Science And Engineering

Time: 3 hours

Max Marks: 80

Answer any FIVE Questions

All Questions carry equal marks

1. (a) Explain the Interrupts of 80286 in the order of priority?
- (b) Explain the salient features of Pentium Processor? [8+8]
2. (a) Explain memory addressing modes of 8086? Give an example for each addressing mode?
- (b) What is the purpose of Trap flag? Discuss how debugging feature is provided with the help of Trap flag in 8086? [8+8]
3. Describe the function of the following pins in 8086 maximum mode of operation. (a) \overline{MN}/MX
- (b) RQ/GT_0 and RQ/GT_1
- (c) QS_0 & QS_1
- ~~(d) LOCK~~ [16]
4. (a) Using REPEAT-UNTIL construct, develop a sequence of 8086 instructions that reads a character string from the keyboard and

after pressing the enter key the character string is to be displayed again.

(b) What is a procedure? Give an example to declare a procedure as near? Make this procedure as PUBLIC procedure? [8+8]

5. (a) Explain the application of stepper motor in microcomputers?

(b) Explain with a neat block diagram the working of dual slope ADC? How do you interface the dual slope ADC to microprocessor? Give the required instruction sequence to acquire one sample from ADC? [8+8]

6. (a) Discuss the following signal descriptions?

—— i. $\overline{INT_0}/INT_1$

ii. TXD

iii. T_0 AND T_1

—— iv. RD

(b) Draw and discuss the formats and bit definitions of the following SFRs in 8051 microcontroller?

i. TMOD

ii. PSW [8+8]

7. In an 8086 based system it is necessary to serve 64 IRQs from different initiators. The allocated address space for 8259s is from 0700h to 070FH. Give the complete design by choosing the appropriate address locations in the above range? Give the initialization sequence for all 8259's with each IRQ activated in level triggered mode and the starting interrupt is type 40H? [16]

8. (a) What are the MODEM control lines? Explain the function of each line?

Discuss how MODEM is controlled using these lines with necessary sequence of instructions?

(b) Interface 8251 with 8086 at address 0F0H. Initialize it in asynchronous mode, with 8 bit character size, baud rate factor 16, one start bit, two stop bits, even parity enable? [8+8]

III B.Tech I Semester Examinations, November 2010

MICROPROCESSORS AND INTERFACING

Common to Information Technology, Instrumentation
And Control Engineering, Electronics And Computer
Engineering, Computer Science And Engineering

Time: 3 hours

Ma

x Marks: 80

Answer any
FIVE Questions

All Questions carry
equal marks

1. (a) Draw the block diagram of 8086 and explain each block.
(b) Discuss the addressing modes provided by 8086 and explain with examples. [8+8]
2. (a) Using DF flag and string instructions, write an assembly language program to move a block of data of length 'N' from source to destination. Assume all possible conditions.
(b) Discuss how procedures are defined and involved in assembly language programming. [8+8]
3. Why do we prefer interrupt driven data transfer than programmed I/O transfer?

Show the complete hardware design to resolve the multiple interrupts based on priority? [16]
4. (a) Explain USB operation?
(b) Interface 8251 with 8086 at address 0A010H. Initialize it in asynchronous mode, with 6 bit character size, baud rate factor 16, one start bit, two stop bits, odd parity enable? [8+8]
5. (a) Explain DOS interrupt 21H and its functions?
(b) Under what conditions type 0 interrupt is initiated? List out the instructions that may cause type 0 interrupt? [8+8]
6. It is necessary to initialize interrupt for mode 2 operation of port-A

and mode

1 operation of port-B with the 8255 address map of 0600H to 0603H. Give the complete hardware design to interface 8255 to 8086 processor with this address map? Write the instruction sequence for the initialization of 8255 in the above modes? Give the instruction sequence to change the operation modes of port A and Port B to mode 1?

[16]

7. (a) Explain the paging system of 80386.

(b) Explain the protected virtual address mode of 80286 and show how 24 bit physical address is generated.

[8+8]

8. Interface two 8255s to 8051 with starting address of 0FFF0H? Show the hardware design? Write the instruction sequence to initialize all ports of 8255s as input ports in mode 0.

[16]

III B.Tech I Semester Examinations, November
2010

MICROPROCESSORS AND
INTERFACING

Common to Information Technology, Instrumentation And
Control Engineering, Electronics And Computer Engineering,
Computer Science And Engineering

Time: 3 hours

Max

Marks: 80

Answer any FIVE
Questions

All Questions carry equal
marks

1. (a) What is a recursive procedure? Write a recursive procedure to calculate the factorial of number N, where N is a two-digit Hex number?

(b) Give the assembly language implementation of the

following. i. REPEAT – UNTIL ii. FOR [8+8]

2. (a) Explain initialization command words and their sequence of operation?

(b) Under what conditions type 0 interrupt is initiated? List out the instructions that may cause type 0 interrupt? [8+8]

3. (a) How do we connect RS-232C equipment

i. To data terminal type devices?

ii. To serial port of SDK 86, RS-232C connection?

(b) Give the specifications of

i. RS-232C

ii. RS-423A [8+8]

4. Discuss the following signal

descriptions? (a) ALE/PROG

(b) EA / VP P

(c) P SEN

(d) RXD

(e) $\overline{\text{INT}}_0$ / INT₁

(f) TXD

(g) T₀ AND T₁

(h) RD [16]

5. 8086 processor do not provide memory indirect addressing mode. Show all possible ways to access a word from memory where the segment address is given in location C000H:1000H and the offset is given in location C000H:1002H. Give the instruction sequence for every addressing mode of 8086? [16]

6. (a) How many local and global descriptors can be defined in 80286 and explain how to access them?
- (b) Discuss the branch prediction logic of Pentium processor? [8+8]
7. What is function of ready pin in 8086. Draw the circuit diagram for wait state generation between 0 and 7 wait status and draw the corresponding timing diagram. [16]
8. Interface an 8-bit DAC to 8255 with an address map of 0804H to 0807H. The DAC provides output in the range of +5V to - 5V. Write the instruction sequence for the following?
- (a) For generating a square wave with a peak to peak voltage of 2V and the frequency will be selected from memory location 'FREQ'.
- (b) For generating a triangular wave with a maximum voltage of +4V and a minimum of -2V.[8+8]

III B.Tech I Semester Examinations, November 2010

MICROPROCESSORS AND INTERFACING

**Common to Information Technology, Instrumentation And
Control Engineering, Electronics And Computer Engineering,
Computer Science And Engineering**

Time: 3 hours

Max

Marks: 80

Answer any FIVE
Questions

All Questions carry equal
marks

1. (a) Write a sequence of instructions to communicate to a modem using 8251 at address 080H.
- (b) Give the specifications of
- i. RS-232C
 - ii. RS-423A
- [8+8]

2. (a) What is the purpose of operational command words of 8259?
Explain their format and the use.

(b) Explain the following terms with reference to 8259.

i. Fully nested mode

ii. Automatic rotation

[8+8]

3. Describe the function of the following pins and their use in 8086 based system.

(a)

— NM~~I~~

(b)

— LOC

K

(c)

T

EST

(d) RESET

[16]

4. (a) Discuss various branch instruction of 8086 microprocessor, that are useful for relocation?

(b) Using a do-while construct, develop a sequence of 8086 instructions that reads a character string from the keyboard and after pressing the enter key the character string is to be displayed again.

[8+8]

5. (a) Draw and discuss the formats and bit definitions of the following SFRs in 8051 microcontroller?

i. SCON

ii. TCON

(b) Discuss the following signal descriptions?

i. ALE/PROG

— ii. EA / V_{pp}

— iii. PSEN

v. RXD

[8+8]

6. It is necessary to check whether the word stored in location 6000H:6000H is zero or not. Show all possible ways of testing the above condition using different address-

ing modes and store 0FFH if the condition is satisfied in location A000H:1002H. Otherwise store 00H. [16]

7.(a) Discuss memory management of virtual 8086 mode in 80386.

(b) Bring out the architectural differences between 80386 and Pentium processors.

8. Interface a stepper motor with 8-step input sequence to 8086 based system and write the instruction sequence to move the stepper motor 20 steps in clockwise and 12 steps in anti-clockwise direction. [16]

III B.Tech II Semester Regular Examinations, April/May 2009

MICROPROCESSORS AND INTERFACING

(Common to Electronics & Communication Engineering, Electronics & Instrumentation Engineering, Bio-Medical Engineering, Electronics & Control Engineering and Electronics & Telematics)

Time: 3 hours

Max Marks: 80

Answer any FIVE Questions

All Questions carry equal marks

1. (a) Draw the architectural diagram of 8085 and explain the function of each block in detail
(b) Discuss about Multiplexing in 8086 microprocessor [10+6]
2. (a) Explain in detail about the following addressing modes of 8086 with examples. i. I/O port addressing
ii. Based indexed addressing with displacement
(b) Write an ALP in 8086 to add two 16-bit hexa decimal numbers [10+6]
3. (a) Write an ALP in 8086 to sort a given set of 8-bit unsigned integers into as-cending order by bubble sort method
(b) Write an ALP in 8086 to display the string "WELCOME" on the screen [10+6]

4. (a) Explain how static RAMs are interfaced to 8086. Give necessary interface diagram assuming appropriate signals and memory size
(b) Explain the need of DMA. Discuss in detail about DMA data transfer Method
5. (a) Distinguish between Mode set control word and BSR control Word of 8255? (b) Write an ALP in 8086 to generate a symmetrical square wave form with 1KHz frequency? Give the necessary circuit setup with a DAC?[8+8]
6. (a) Draw the internal block diagram of 8259, and explain in detail about each block?
(b) With examples, discuss briefly about software and hardware interrupts. [10+6]
7. (a) Discuss the types of serial communication?
(b) Write an 8086 instruction sequence for receiving 50 characters using 8251 and store them in memory at location 2080H.. [8+8]
8. (a) Discuss about various addressing modes of 8051.
(b) Explain the interrupt structure of 8051 [8+8]

III B.Tech II Semester Regular Examinations, April/May
2009

MICROPROCESSORS AND INTERFACING

(Common to Electronics & Communication Engineering, Electronics
& Instrumentation Engineering, Bio-Medical Engineering, Electronics
& Control Engineering and Electronics & Telematics)

Time: 3 hours

Max Marks: 80

Answer any FIVE Questions

All Questions carry equal
marks

1. Explain the function of following registers in 8086 microprocessor : [4×4=16]

(a) AX,BX,CX,DX

(b) CS,DS,SS,ES

(c) BP,SP,SI,DI

(d) IP and Instruction Queue

2. (a) Explain the following 8086 instructions with
examples. i. MUL

ii. IMUL

iii. DIV

iv. IDIV

(b) Differentiate between procedures and macros using certain examples.

[8+8]

3. (a) Write a program in 8086 to add two 8-bytes of data available in memory location array1 and array 2. Store the result in array3
(b) Write an ALP to count number of 0s in a 16 bit binary string [10+6]
4. (a) With a neat block diagram, explain the working of 8257 DMA controller
(b) Explain briefly about memory interfacing with 8086 microprocessor [10+6]
5. (a) Write an ALP in 8086 to generate triangular waveform and give the necessary circuit setup with a DAC.
(b) Write an algorithm for driving a stepper motor. Assume that the desired direction stored in BL and the number of steps is stored in CL. Write a delay routine for 1 millisecond after each step movement. [8+8]
6. (a) Distinguish between interrupts and exceptional handling interrupts
(b) Discuss about the following terms:
i. Vector interrupt table and lookup table.
ii. Interrupt service routines and subroutines. [8+8]
7. (a) Discuss the types of serial communication?
(b) Write an 8086 instruction sequence for receiving 50 characters using 8251 and store them in memory at location 2080H.. [8+8]
8. (a) Explain the internal and external program memory as well as data memory of 8051 with the diagram showing their capacities.
(b) Draw the diagram to Interface Program memory of 16K x 8 EPROM to 8051 and give its memory map. The address of memory map should start from 0000H. [8+8]

III B.Tech II Semester Regular Examinations, April/May 2009

MICROPROCESSORS AND INTERFACING

(Common to Electronics & Communication Engineering, Electronics & Instrumentation Engineering, Bio-Medical Engineering, Electronics & Control Engineering and Electronics & Telematics)

Time: 3 hours

Max Marks: 80

Answer any FIVE Questions

All Questions carry equal marks

1. (a) Compare the Interrupt structure of 8086 and 8085 Microprocessors.
Also discuss about priorities of the interrupts in both the cases.
(b) With examples ,explain how multiplexing is implemented in 8086 Microproces- sor [10+6]
2. (a) Explain in detail about the following addressing modes of 8086 with examples. i. I/O port addressing
ii. Based indexed addressing with displacement
(b) Write an ALP in 8086 to add two 16- bit hexa decimal numbers [10+6]
3. (a) Write an ALP in 8086 to count number of positive and negative numbers from an array of 8-bit integers
(b) Write an ALP in 8086 to exchange a block of N bytes of data between source and destination [8+8]
4. (a) With relevant pin diagrams explain the minimum and maximum mode oper- ations of 8086
(b) Explain briefly about DMA data transfer method. [12+4]
5. (a) With a neat internal architectural diagram, explain the features of 8255?

(b) Explain the interfacing of a stepper motor with 8086 using the ports of 8255?
[8+8]

6. (a) What is the purpose of operational command words of 8259? Explain their format and the use?

(b) Discuss the following
interrupts. i. Divide Error
ii. Non Maskable interrupt
iii. Break Point interrupt

[10+6]

7. (a) Write the instruction sequence in 8086 to initialize the 8251

(b) Draw and explain the block diagram of 8251. [8+8]

8. (a) Draw the architectural diagram of 8051 microcontroller and explain in detail about each block.

(b) Explain the basic differences between a microprocessor and a microcontroller.

[10+6]

17.Question bank:

UNIT-1

General definitions of mini computers etc.,

1. What are the advantages and the limitations microcontroller over a Microprocessor?
2. Describe the main blocks in a digital signal processor that are not in a general Microprocessor?
3. Overview of 8085 Microprocessor
4. List the internal registers in 8085 microprocessor and their abbreviations and lengths. Describe the primary function of each register.
5. Differentiate between NMI and MI interrupts
6. Explain how with external hardware TRAP can be masked
7. Explain the timing diagrams of 8085 when it is executing Memory mapped I/O and I/O mapped I/O instructions
 - B) Explain the following pins of 8085 microprocessor
 - i) SID,SOD ii) ALE iii) HOLD,HLD iv) TRAP
- 8.Explain the features of 8086
9. Explain architecture if 8086
10. Explain the function of following registers in 8086 microprocessor.
 - a) AX, BX, CX, DX. b) CS, DS, SS, ES. C) BP, SP, SI ,DI.
 - d) IP & instruction queue e) Flags.

UNIT-2

1. List out the addressing modes of 8085, & Explain each with an example
2. Write a algorithm & ALP to reverse the bits in a 16-bit number & check weather it is a Palindrome or not.
- 3.Explain all branch and call instructions

4. Write an ALP for Ascending and Descending of a series
5. Write an ALP for Matrix multiplication
6. Explain the meaning of the following 8086 instructions.
 - i) MOV [3845H],BX ii) ADD AX,[SI]
 - iii) MOV BX,2956H iv) ADC AX,BX.
7. Explain string instructions supported by 8086 processor.
8. Write a algorithm & ALP to reverse the bits in a 16-bit number & check weather it is a Palindrome or not.
9. If BH = 0F3H what is the value of BH in hex after the instruction SAR BH, 1
10. IF AL = 78H and BL=73H explain how DAS instruction (after subtracting BL From AL) adjusts to the BCD result
11. If CL=78H what is the value of CL after the instruction ROL CL, 3
12. Why AAD is to be executed before DIV instruction while converting unpacked BCD to Binary number
13. Under what conditions REPE MOVS executes
14. Explain XLAT instruction to linearize transducer characteristics
15. Explain intra segment and inter segment branch instructions with examples the instructions related to arithmetic and logical shift.
16. Explain all addressing modes with the assembler syntax and how effective address is calculated

UNIT-3

1. How do you configure 8086 into minimum and maximum modes
2. Bring out the differences between 8086 and 8088 processors
3. Explain all the features in 8284
4. Why and when wait states are required. How do you insert wait states1.

5. . In an SDK-86 kit **128KB SRAM** & **64KB EPROM** is provided on system and provision Expansion of another **128KB SRAM** is given. Then on system SRAM address starts from **00000H** & EPROM ends with **FFFFFH**. The expansion slot address map is from **80000H** To **9FFFFH**. The size of **SRAM chip is 64KB, EPROM chip size is 16KB**. Give the Complete memory interface and also the address map for complete memory map for Individual chips..
6. What are the steps in interfacing peripherals with the micro processor?
7. Sketch and explain the interface of PPI 8255 to the 8086 microprocessor in minimum mode. Interface 4 7 segment LEDs to display as a BCD counter
8. In the above question Q2 interface two keys UP and DOWN to the PPI. Write an 8086 assembly program segment such that when UP is pressed the counter counts up every second. Similarly when DOWN key is pressed the counter decrements every second
9. Sketch and explain the interface of 8279 to the 8086 microprocessor in minimum mode. Interface 8x8 key pad and 16x 7 Seg LED display. Write an 8086 assembly program to read the key codes of keys and display -NPTEL-INDIA
10. Sketch and explain the interface of PIT 8254 to the 8086 microprocessor in minimum mode. Cascade two counters in the PIT. Write a program segment two get one minute delay
- 11 What are the differences in interfacing RWMs while 8086 is in minimum and maximum modes?
12. Sketch and explain the interface of 32K x 16 RWMs using a decoder in minimum mode. What is the maximum access time of ROMs such that it does not require wait states when 8086 operates at 8 MHz?
13. Sketch and explain the timing diagrams in the above interface Question 2

- 14 Sketch and explain the 8086 bus activities during write machine cycle
15. Review on Interrupt Structure of 8086
16. Explain serial data transfer schemes
17. Pin configuration of 8251
18. Architectural features of 8251
19. Sending and receiving a character

UNIT-4

1. What are the advantages and disadvantages of using Harvard architecture in 8051?
2. How much maximum external program memory can be interfaced?
3. Explain PSW SFR. Give the application differences between Carry and Overflow flags
4. What are the power consumptions in power down and idle modes
5. Explain Quasi Bidirectional ports of 8051
6. What is the status of all registers on reset?
7. What is the maximum delay the Timer0 produces when 8051 is operated at 12MHz?
8. Explain how in Serial communication mode 0 expands I/O lines with the help of Shift

UNIT-5

1. explain timers/counters?
2. explain different modes of timers/counters?
3. explain serial communication concepts in 8051?
4. explain different modes in serial communication?
5. explain interrupt structure of 8051?

18. Assignment Questions

UNIT 1

1. Explain pin description of 8086 with functional pin diagram.
2. Draw the 8086 architecture diagram and explain various blocks of the diagram. Detail the Register organization in 8086.
3. What is memory segmentation? Describe physical memory organization. How to calculate the physical memory address.
4. Describe minimum mode and maximum mode operation in 8086 along with Timing diagrams.
5. Explain programmers model and flag register format of 8086

UNIT 2

1. Explain the instruction formats in 8086. give the complete instruction set of 8086 along with examples.
2. What is an addressing mode? Explain various addressing modes of 8086 along with examples.
3. What is an assembler directive and Macro? Give the differences between Procedure and Macro. Explain Assembler directives, Procedures and Macros with the help of examples.
4. Write ALP for
 - a. to perform multi-byte addition.
 - b. to perform the sorting N numbers.
 - c. to display the string "GEETHANJALI"
 - d. to move a block of data from source to destination.
 - e. to evaluate the expression $s = uv + v^2$

UNIT 3

1. Explain 8255 architecture and various modes of data transfer.
2. Keyboard and display interfacing
3. ADC and DAC interfacing
4. Stepper motor interfacing

5. Explain how memory is interfaced with 8086. Write memory connection diagrams for minimum mode and maximum mode operation of 8086.
6. Explain interrupt structure of 8086. What is vector table? what are the operations done during handling an interrupt service routine?
7. Explain why serial data transfer is mostly preferred over parallel data transfer. Give reasons? Discuss the types of serial communication?
8. Draw the internal block diagram of 8251 USART and explain in detail about each block.
9. With a neat circuit diagram, explain the interfacing of 8251 with 8086
10. Design a hardware interfacing circuit for interfacing 8251 with 8086. Set the 8251A in asynchronous mode as a transmitter and receiver with even parity enabled, 2 stop bits, 8-bit character length, frequency 160 KHz and baud rate 10K. Write an ALP to transmit 100 bytes of data string starting at location 2000:5000H.
11. Give the description about the following.
 - RS-232 standard
 - Ttl to rs232 conversion
 - Rs232 to ttl conversion.

UNIT-4

1. 8051 microcontroller architecture
2. Register organization of 8051
3. Addressing modes and i/o ports of 8051.
4. Memory organisation of 8051.
5. Operation of I/O ports.

UNIT-5

1. programming Timer/counter and Various modes of operation of timers.
2. Serial communication operation
3. Memory interfacing
4. Programming external hardware interrupts

19. Unit-wise quiz questions

-----flag is used in 8086 for string manipulation instructions

A DF

B AF

C OF

D

8086 as ----- of segment registers

A 3

B 5

C 4

D 2

A is an identifier that is assigned the address of the first byte of the instruction in which it appears

A Operand

B mnemonic

Label

D Comment

In which instruction the flags are not affected

A LDS

B DAA

C ADC

D DAS

Data storage in stack is designed in _____method.

First in first
out

B last in last out

first in last out

last in first out

----- instruction is used to call a procedure

A CALL

B RET

C PROC

D NEAR

-----is a prefix used in 8086 machine language that simplifies the use of string primitives with loops

REP

B LOOP

C LODS

D MOVS

LODSB instruction will automatically increment _____register.

A

DI

B

SI

C

DS

D

ES

----- is a special component used in 8086 for block transfer of data to or from a port

A

BMA

B

DNA

C

The first operand in the IN instruction must be -----register for byte transfer

A CL

B BL

AL

D DL

In _____ environment two or more process is executed in a time multiplexed fashion

A Uniprogramming

B Multiprogramming

C I/O operation

D DMA

Preventing two or more process from simultaneously entering their critical sections for accessing a shared resource is called-----

A Mutual sharing

B Mutual process

Mutual exclusion

D mutual accessing

.....IC is used as clock generator for 8086

A 8087

B 8284

C 8088

D 8294

A processor is in minimum mode when its MN/MX pin is strapped to -----volts

A +50

B +5

C 0

D -5

8086 is interfaced with -----Co-processor to perform float point calculations.

- A 8089
- B 8085
- C 8087
- D 8084

An instruction to be executed by the coprocessor is indicated when -----instruction appears in the program sequence

- A TRUE
- B FALSE
- C TEST
- D ESC

One of the following addressing modes is not possible in 8085.

- A Indexed addressing
- B Indirect addressing
- C Direct addressing
- D Indirect register address

The speed of a microprocessor is usually measured by the

- A Microprocessor's throughput.
- B Speed with which it performs I/P and O/P operations.
- C Time required executing a basic instruction.
- D Time required processing a small operation

Which of the following is an 8085 hardware interrupt?

- A TRAP,
- B RST6.5
- C RST7.5,
- D All the above

Which of the following does NOT belong to 8085 flag register

- A Carry flag
- B Sign flag
- C Zero flag
- D One flag

DAD re instruction mean

- A Add the content of specified reg. to content of HL & store the result in HL
- B Add 8 bit data given in inst to content of one & store the result in reg
- C Add 10 bit data given in inst to content of HL & store the result in HL
- None of the above
- Instruction that sets the carry flag
- D A CY set
- B STR

C STC

D Set CY

Instruction that avoids unnecessary jumps to terminate the prog

A NOP

HLT

C ei

D O

Instruction cycle can be divided as sum of

A Instruction fetch + Instruction decode

B Instruction fetch + instruction Execute

C Instruction fetch + instruction exit

D All of above

Instruction cycle use _____ m/c cycle,

A 1-5

B 1-16

C 1-8

D None of above

Each m/c cycles have _____ T states

A 3 to 6t

B 3 to 10t

C 3 to 8t

D None of above

Which of the following does the task of displaying the result computed by the microprocessor?

A Cathode Ray Tube (CRT)

B Light-Emitting diodes (LED'S)

C Laser printer

D All of the above.

In 8086 microprocessor which of the following is Not a register pair register

A B-C pair

F-G pair

C H-L pair

D D-E pair

Zero Flag is to _____ if the result of an arithmetic operation is zero.

1

B 0

C -
1

D 2

Two's complement representation of -2 is _____

A 0010

B 0110

C 1110

D 0011

If there are two operands, the _____ operand appears first.

A destination

B Destination

C source

D labe

An interrupt initiated by a signal on the NMI pin is called a _____

A NON maskable interrupt

B Number mask able interrupt

C NOT mask able interrupt

D None of the abov

There are two power connection for +5 volts in 8085 they are

A Vss and Vgg

Vcc and Vss

C Vcc and Vgg

D None of the above

Which of the following is the function of microprocessor

A Receiving input

B Performing computations.

C Storing data & instructions

D All of the above.

----- and ----- lines provide the 8085 with a DMA capability by allowing another processor on the same system bus to request control of the buses

A S0,S1

B Read and Write

C ALE and READY

D HOLD and HLDA

-----and ----- instructions provide a means of moving two bytes between HL register and a RAM address

A LDAX & STAX

B LHLD & SHLD

C LDA& STA

D PUSH & POP

The binary representation of decimal number 34.56is

A 100011.100

B 100010.110

C 100010.100

D 100011.111

The octal representation of hexadecimal number CD is

A 315

B 305

C 335

D 312

8086 processor has ----- address pins out of which ----- number of pins are used as data pins

A 16,8

B 16,14

C 20,16

D 20,8

In 8086 is a -----bit microprocessor and is fabricated using -----technology

A 16,CMOS

B 8,HMOS

C 20,NMOS

D 16,HMOS

_____ and _____ instructions are specifically used to change the status of carry flag

A STI,CLI

B STC,CLC

C MOV,LEA

D DAA,XCHG

In 8086 theinstruction is used to simplify the decrementing, testing and branching. This instruction uses _____as Counter register.

Loop, CX

B JMP, CX

C DEC, AX

D INC, AX

Two general class of interrupt are _____ and _____

A Pointer and vector

B IN and OUT

C input and output

D internal ,external

A _____is a segment of code that needs to be written only once but whose basic structure can be caused to be repeated several times within a source module but placing a single statement at the point of each appearance

A Micro

B Macro

C prototype

D interrupt

Packed BCD format of number 01 02 04 05 is -----and unpacked BCD format of Number 5789 is-----

A 10203040, 50 70 80 09

B 11223344, 55 67 78 99

C 1234, 05 07 08 09

D 10203040,05 07 08 09

In 8086 _____instruction is specially designed for executing procedure .this instruction assumes the base address of the byte array is in BX register and the byte to be converted is in the _____register

A ASCII, CX

B EBCDIC, CX

C PCHL, AL

D XLAT, AL

In 8086 all programmed communication with the I/O ports is done by the ----- and ----- - instructions

A MOV, OUT

B LOAD, MOVE

C IN, OUT

D IN, LOAD

In 8086 the external interrupts is caused by a signal sent to CPU through one of its pins they are _____and _____.

A NMI, IRET

B INTR, RET

C NMI, INTR

D RET,IRET

In _____ memory management system fragmentation problem appear. Fragmentation problem can be eliminated by-----allocation scheme

A Partition allocation, relocation partition

B virtual memory, relocation partition

C dynamic ,partition

D relocation partition, dynamic

57.8086 can be operated in Two modes they are -----and-----

A Minimum, Maximum

B External, internal

C Mode1, Mode2

D Data, address

-----IC is designed specifically to work with 8086 as programmable interrupt controller, it contains-----number of pins

A 8259,28

- B 8269,28
- C 8279,26
- D 8299,26

The 8087 internally stores all numbers in the ----- format which uses 15 bits for the exponent and ----- bits for the mantissa

- A permanent real,32
- B **temporary real,64**
- C permanent integer,32
- D temporary integer,64

8087 numeric data processor is specially designed to perform arithmetic operations it can operate on data with length ranging from ----- to -----

- A 2, 6
- B **2, 10**
- C 4, 8
- D 4, 10

Interrupt type is given a number between _____ and _____ and the address of the interrupt pointer is found by multiplying the type by _____ if type is 9 then the interrupt pointer will be in bytes-----through -----

- A 0 and 256, 5, 00025 through 00030
- B 0 and 200, 4, 00020 through 00024
- C 0 and 200,4, 00024 through 00028
- D **0 and 255,4,00024 through 00027**

The 8086 receives bus request through its pin and issues grants from its pin .A request is made when a potential master sends to pin

- A **HOLD,HLDA,1,HOLD**
- B HLDA,HOLD,1,HLDA
- C HOLD,HLDA,0,HOLD
- D HLDA,HOLD,0,HLDA

Intel IC 8286 transceiver device contains ----- tristate elements, -----receivers and -----drivers. In order to operate 8086 in minimum mode -----number of 8286ICs are needed.

- A 8,4,4,2
- B 10,5,5,1
- C 6,3,3,1
- D **16,8,8,2**

The binary representation of hexadecimal number ABC is

10101011110
0

101010111000
011010111100

10101011111

Subtracting octal number 478 from octal number 568 results in

5

7

6

8

8085 micro processor is an ----- device

PMOS

CMOS

NMOS

QMOS

----- register deals with sequencing the execution of instructions

Stack pointer

program counter

accumulator

flag

1. Mode 1 of 8255 is used for which of these I/O methods

(a) Bi-Directional Handshake

(b) DMA I/O

(c) **Strobed or Hand Shake mode**

(d) Simple Input/Output

2 . If the 8255 is selected for addresses 0F800H to 0F806H, which one of the following is the address of PORT C

(a) 0F806H

(b) 0F802H

(c) **0F804H**

(d) 0F800H

3 . If 8 seven segment led multiplexed displays including the decimal point are connected. How many wires are needed to connect the segments (not including power supply wires)?

(a) 15

(b) **16**

(c) 56

(d) 64

4 . The diode kept across the coil of a step per motor is for ?

(a) **Shorting the Back EMF**

(b) To stop the step per motor

(c) Stopping flow of current

(d) Allowing flow of current

5 . In the A/D converter ADC0808 which of these signals tri states the digital output of ADC

(a) EOC

- (b) ALE
- (c) SOC
- (d) OE**

6 . The fourth response of the 8086 to an interrupt is which one of the following ?

- (a) Decrements the SP by 2 & pushes flag register
- (b) It disables interrupt flag
- (c) Decrements the SP& pushes current CS contents
- (d) Resets Trap Flag**

7 . What is the size of the vector interrupt table?

- (a) 256 bytes
- (b) 65536 bytes
- (c) 1024 bytes**
- (d) 4096 bytes

8 . Which of these values in AH perform input from keyboard ?

- (a) 02 (b) 09 (c) 17 (**d) 01**

9 . Which one of these BIOS services is used for Mouse Operation?

- (a) 10H
- (b) 13H
- (c) 16H
- (d) 33H**

10. The PIC interrupt controller which Masks the Interrupt is?

- (a) IRR
- (b) IMR**
- (c) ISR
- (d) PR

11. Data line which can be transfer in only one direction is referred as

- (a) Half Duplex
- (b) Com plex
- (c) Full Duplex
- (d) Simplex**

12. In the Synchronous mode the 8251 which signal indicates that the specified Synchronous character is received?

- (a) Tx C
- (b) SYNDET/BD**
- (c) CLK
- (d) C/D

13. The outer loop counter for N numbers is

- (a) N-1** (b) N-2 (c) N (d) N/2

14. If the CPU has not read a character from the 8251 before the arrival of the next character. Which of the following errors is supposed to have occurred?

- (a) Even Parity error
- (b) Framing Error
- (c) Over Run Error**
- (d) Odd Parity Error

15. USB stands for which one of the following?

(a) Universal Sequential Bus

(b) Universal Serial Bus

(c) College Bus

(d) Universal Sequential Bus

16. The 8051 is a

(a) 32 bit micro controller

(b) 24 bit micro controller

(c) 16 bit micro controller

(d) 8 bit micro controller

17. If the bit 8th bit of memory location 20H is to be set, what will be its bit addressable memory location?

(a) 09

(b) 07

(c) 06

(d) 08

18. Counter or timer operation is chosen in which of these registers?

(a) TLx

(b) TCON

(c) THx

(d) TMOD

19. Serial port operates in Multi Processor mode in which of these modes?

(a) Mode3

(b) Mode2

(c) Mode0

(d) Mode1

20. The upper 8 bits of address bus are generated on which of these ports?

(a) Port2

(b) Port3

(c) Port1

(d) Port0

21 . Mode 1 of 8255 is used for which of these I/O methods

(a) Simple Input/ Output

(b) DMA I/O

(c) Bi-Directional Hand shake

(d) Strobed or Hand Shake mode

22 . Two 8255's IC1&IC2 are connected to 8086 16 bit bus and D0 to D7 are connected to IC1 and D8 to D15 are connected to IC2. The base address signals A0&A1 goto IC1&IC2's A0&A1 pins respectively. If a decoded signal with a range of 0FFE0H to 0FFE3H then What is the address of IC1PORT B register

(a) 0FFE2H

(b) 0FFE0H

(c) 0FFE3H

(d) 0FFE1H

23 . If 8 seven segment led multiplexed displays including the decimal point are all lighted up and each of these is biased to carry 5mA. What is the total current drawn from the power supply when all the segments are ON?

(a) 280 mA

(b) 80 mA

(c) 40 mA

(d) 300 mA

24 . The diode kept across the coil of a stepper motor is for?

(a) Stopping flow of current

(b) To stop the stepper motor

(c) Allowing flow of current

(d) Shorting the Back EMF

25 . In the A/D converter ADC 0808 which of these signals tri states the digital output of ADC

(a) OE

(b) SOC

(c) EOC

(d) ALE

..

..

26 . The fourth response of the 8086 to an interrupt is which one of the following?

(a) Decrements the SP by 2 & pushes flag register

(b) Resets Trap Flag

(c) It disables interrupt flag

(d) Decrements the SP & pushes current CS contents

27 . What is the segment address of the vector interrupt table?

(a) 0F0000H

(b) 00000H

(c) 10000H

(d) 0F000H

28 . Which of these values in AH performs output on video display of Ascii character in DL?

(a) 01

(b) 09

(c) 02

(d) 17

29 . Which one of these BIOS services is used for Key Board Operation ?

(a) 33H

(b) 13H

(c) 16H

(d) 10H

30. The PIC interrupt controller connects the single 8086 IRQ to how many IRQ input lines?

(a) 4

(b) 8

(c) 32

(d) 16

31. Data line which can be transfer in only one direction is referred as as

(a) Full Duplex

(b) Half Duplex

(c) Complex

(d) Simplex

32. USART stands for which of the following?

(a) Unidirectional Synchronous Asynchronous Receiver Transmitter

(b) Unidirectional Sequential Asynchronous Receiver Transmitter

(c) Universal Synchronous Asynchronous Receiver Transmitter

(d) Universal Serial Addressed Receiver Transmitter

33. Which combination of the flags indicate that the source is greater than the destination

(a) CY=1,Z=1

(b) CY=0,Z=0

(c) CY=0,Z=1

(d) CY=1,Z=0

34. When a valid stop bit is not detected at the end of every character in a synchronous mode, the error is called as which one of the following?

(a) Over Run Error

(b) Odd Parity Error

(c) Even Parity error

(d) Framing Error

35. USB stands for which one of the following?

(a) Universal Sequential Bus

(b) Unidirectional Serial Bus

(c) Universal Serial Bus

(d) SMCE bus

36. The size of the ROM in 8051 is

(a) 32 K bytes

(b) 16 K bytes

(c) 4 K bytes

(d) 8 K bytes

37. If the bit 8th bit of memory location 20H is to be set, what will be its bit addressable memory location?

(a) 06

(b) 09

(c) 08

(d) 07

38. The TF bit is in which of these registers?

(a) TCON

(b) TLx

- (c) THx
- (d) TMOD

39. The interrupt vector address for timer 0 is which one of the following addresses?

- (a) **000BH**
- (b) 0003H
- (c) 0023H
- (d) 0013H

40. The instruction used to access data from external memory is which one of the following?

- (a) MOV @
- (b) MOV
- (c) MOVC
- (d) **MOV X**

41 . The number of ports 8255 PPI has is

- (a) 1 (**b**) **3** (c) 2 (d) 4

42 . If the control word 09BH is given to the control register of the 8255 PPI then which of these options are the condition of the ports

- (a) PORT's A&B INPUT,PORT C OUTPUT
- (b) **ALL PORT's INPUT**
- (c) BIT SET RESET
- (d) ALL PORT's OUTPUT

43 . A key board is made using 12 keys connected not as a matrix. How many pins of the I/O port are used?

- (a) **12**
- (b) 07
- (c) 11
- (d) 09

44 . What is the angle per step of a step per motor in full step mode?

- (a) 1.0 degrees
- (b) **1.8 degrees**
- (c) 0.45 degrees
- (d) 0.9 degrees

45 . In the A/D converter ADC 0808 which of these signals is used select an analog input channel to be digit is used

- (a) **ALE**
- (b) SOC
- (c) OE
- (d) EOC

46 . The second response of the 8086 to an interrupt is which one of the following?

- (a) Resets Trap Flag
- (b) Decrements the SP by 2&pushes flag register
- (c) Decrements the SP & pushes current CS contents
- (d) **It disables interrupt flag**

47 . Which of these is called when an NMI occurs?

- (a) INT3
- (b) INT2**
- (c) INT0
- (d) INT1

48 . The DOS function AH=9 INT2 1H displays a message till which one of this character is reached

- (a) \$ (b) \times (c) # (d) ***

49 . Which one of these BIOS services is used for Communication Operation?

- (a) 17H
- (b) 15H
- (c) 16H
- (d) 14H**

50. The total number of interrupts available if all the 8 inputs of PIC are each having a slave is?

- (a) 128
- (b) 64**
- (c) 256
- (d) 16

..
..

51. The USART used as Serial Communication controller is

- (a) 8259
- (b) 8257
- (c) 8250
- (d) 8251**

52. USART stands for which of the following ?

- (a) Unidirectional Synchronous Asynchronous Receiver Transmitter
- (b) Universal Synchronous Asynchronous Receiver Transmitter**
- (c) Unidirectional Sequential Asynchronous Receiver Transmitter
- (d) Universal Serial Addressed Receiver Transmitter

53. Given the program shown below , What is the expression evaluated by this program?

```
P db 5
Q db 6
R db 7
MOV AL,[P]
MUL AL
MOV BL,AL
MOV AL,[Q]
MUL AL
ADD AL,BL
MOV [R],AL
```

- (a) $R = P + Q^2$
- (b) $R = (P + Q)^2$**

(c) **R=(P2+ Q2)**

(d) $R=(P+Q)$

54.If the CPU has not read a character from the 8251 before the arrival of the next character. Which of the following errors is supposed to have occurred?

(a) Even Parity error

(b) Odd Parity Error

(c) Framing Error

(d) **Over Run Error**

55. Which one of these serial data transmission standards is single ended and does not use low impedance drivers?

(a) RS449

(b) **RS232C**

(c) RS423A

(d) RS422A

56. Which one of the following is the 8051 architecture based upon?

(a) Princeton Architecture

(b) Param Architecture

(c) **Harvard Architecture**

(d) Von Neumann Architecture

57. General purpose registers R0 to R7 can be referred as sets of four banks (0to3) and also by memory location. If a register R3 in Bank2 is referred, what is its internal ram location address?

(a) 14H

(b) 10H

(c) 12H

(d) **13H**

58. Timer0 is functional as an 8 bit counter while timer 1 is stopped in which of the following modes?

(a) Mode2

(b) Mode1

(c) **Mode3**

(d) Mode0

59. Which of the following is the correct order of priority in decreasing order from left to right?

(a) EX0,EX1,ET0,ET1,ES,

(b) ES,EX1,ET0,E01,ET1

(c) ET0,EX0,ET1,EX1,ES

(d) **EX0,ET0,EX1,ET1,ES**

60. One of these signals indicates that address is available on Port0?

(a) EA

(b) RESET

(c) **ALE**

(d) PSEN

61 . The 8255 PPI has

- (a) 28 pins
- (b) 16 pins
- (c) 40 pins**
- (d) 64 pins

62 . Two 8255's IC1&IC2 are connected to 8086 16 bit bus and D0 to D7 are connected to IC1 and D8 to D15 are connected to IC2. The base address signals A0&A1 goto IC1&IC2's A0&A1 pins respectively. If a decoded signal with a range of 0FFE0H to 0FFE3H then What is the address of IC1 PORTB register

- (a) 0FFE2H
- (b) 0FFE3H
- (c) 0FFE1H**
- (d) 0FFE0H

63 . A key board is made using 12 keys connected not as a matrix. How many pins of the I/O port are used?

- (a) 12**
- (b) 07
- (c) 09
- (d) 11

64 . How many wires does a stepper motor effectively have?

- (a) 4
- (b) 2
- (c) 3
- (d) 5**

65 . An A/D converter which has a maximum output digital value of 1023 need show many wires for connecting its digital output to a peripheral chip

- (a) 10**
- (b) 13
- (c) 8
- (d) 12

66 . Which of these signals has the highest priority?

- (a) IRQ
- (b) Hold
- (c) NMI
- (d) Reset**

67 . What is the number of Software interrupts available in the Interrupt Vector Table?

- (a) 4096
- (b) 256**
- (c) 1024
- (d) 512

68 . DOS is an abbreviation which stands for?

- (a) Data Output Storage
- (b) Disk Operating Systems**
- (c) Disk Output Storage
- (d) Device Operating System

69 . Which one of these BIOS services is used for Mouse

Operation?

(a) **33H**

(b) 13H

(c) 16H

(d) 10H

70. The PIC interrupt controller which indicates that a Interrupt is being requested is?

(a) ISR (b) **IRR** (c) IMR (d) PR

71. The device from which data originates or terminates is called

(a) DSR

(b) DTR

(c) **DTE**

(d) DCE

72. USART stands for which of the following ?

(a) **Unidirectional Synchronous Asynchronous Receiver Transmitter**

(b) Universal Synchronous Asynchronous Receiver Transmitter

..

..

(c) Unidirectional Sequential Asynchronous Receiver Transmitter

(d) Universal Serial Addressed Receiver Transmitter

73. Given the program shown below, What is the value of R after this program executes?

P db 5

Q db 6

R db 7

MOV AL,[P]

MUL AL

MOV BL,AL

MOV AL,[Q]

ADD AL,BL

MOV [R],AL

(a) **R=(P2+Q1)[P2+Q]**

(b) R=(P+Q)

(c) R=(P+Q)2

(d) R=P+Q2

74. When a valid stop bit is not detected at the end of every character in asynchronous mode, the error is called as which one of the following?

(a) Odd Parity Error

(b) Over Run Error

(c) Even Parity error

(d) **Framing Error**

75. USB stands for which one of the following?

(a) Universal Sequential Bus

(b) Uni directional Serial Bus

(c) Universal Serial Bus

(d) Universal Sequential Bus

76. The 8051 micro controller does not have one of the following as its built in peripheral?

(a) Timers

(b) Counters

(c) UART

(d) DMA controller

77. Which of these registers cannot be decremented?

(a) SP

(b) B register

(c) Accumulator A

(d) DPTR

78. One of these registers is loaded with the upper byte of the terminal count?

(a) THx

(b) TMOD

(c) TLx

(d) TCON

79. The bit which disables reception of serial data is which one of the following?

(a) RB8

(b) TB8

(c) RI

(d) REN

80. The instruction which activates the PSEN signal is which one of the following?

(a) MOVC

(b) MOV @

(c) MOV

(d) MOV X

81 .Which of these is Double Hand Shake

(a) STB followed by ACK

(b) STB followed by ACK and removal of ACK

(c) STB only

(d) ACK followed by STB

82 . Two 8255's IC1&IC2 are connected to 8086 16 bit bus and D0 to D7 are connected to IC1 and D8 to D15 are connected to IC2. The base address signals A0&A1 goto IC1&IC2's A0&A1 pins respectively. If a decoded signal with arrange of 0FFE0H to 0FFE3H then What is the address of IC1 PORTB register

(a) 0FFE2H

(b) 0FFE0H

(c) 0FFE3H

(d) 0FFE1H

83 . A key board is made using 12 keys connected as a matrix of 4X3. How many pins of the I/O port are used?

(a) 12 (b) 09 (**c**) **07** (d) 11

84 .How many steps per revolution does a stepper motor in full step mode take?

(**a**) **200**

(b) 400

(c) 380

(d) 220

85 . A n A/D converter which has a maximum output digital value of 1023 need show many wires for connecting its digital output to a peripheral chip

(a) 12 (**b**) **10** (c) 8 (d) 13

86 . Which of these signals has the second highest priority?

(**a**) **NMI**

(b) Hold

(c) Reset

(d) Irq

87 . Which of these is called for doing Single Stepping?

(a) INT0

(b) INT2

(c) INT3

(**d**) **INT1**

88 . Which of these values in AH perform input from key board?

(a) 09 (b) 02 (c) 17 (**d**) **01**

89 . Which one of these BIOS services is used for Video Operation?

(a) 13H (**b**) **10H** (c) 33H (d) 16H

90. The PIC interrupt controller which Masks the Interrupt is?

(a) PR

(b) IRR

(**c**) **IMR**

(d) ISR

91. The device from which data originates or terminates is called

(a) DSR (b) DTR (**c**) **DTE** (d) DCE

92. In the Synchronous mode the 8251 which signal indicates that the specified Synchronous character is received?

(a) CLK

(b) C/D

(c) TxC

(**d**) **SYNDET/BD**

93. Given the program shown below , What is the value or R after this program executes?

P db 5

Q db 6

R db 7

MOV AL,[P]

MUL AL

MOV BL,AL
MOV AL,[Q]
ADD AL,BL
MOV [R],AL

- (a) $R = (P+Q)$
- (b) $R = (P+Q)^2$
- (c) **$R = (P^2+Q^1)[P^2+Q]$**
- (d) $R = P + Q^2$

94. When a valid stop bit is not detected at the end of every character in asynchronous mode, the error is called as which one of the following?

- (a) Over Run Error
- (b) Even Parity error
- (c) **Framing Error**
- (d) Odd Parity Error

95. USB connector has how many pins?

- (a) 3 (b) 2 (c) 5 (d) **4**

96. The 8051 micro controller is?

- (a) **8 bits**
- (b) 32 bits
- (c) 16 bits
- (d) 24 bits

97. Which of these registers cannot be decremented?

- (a) SP
- (b) Accumulator A
- (c) **DPTR**
- (d) B register

98. Which one of the following starts the timer?

- (a) IE1
- (b) TF
- (c) **TR**
- (d) IT1

99. The interrupt vector address for serial reception is which one of the following addresses?

- (a) 0003H
- (b) **0023H**
- (c) 000B H
- (d) 0013H

100. The instruction used to access internal RAM is which one of the following?

- (a) MOV @
- (b) **MOV**
- (c) MOV X
- (d) MOVC

101. The number of ports 8255 PPI has is

- (a) **3** (b) 2 (c) 1 (d) 4

102. When 8255 port is to be read then which of these combinations is valid

- (a) RD= 1 , W R= 0, R E S E T =0 , C S =0
- (b) RD= 0 , W R= 1, R E S E T =1 , C S =0
- (c) R D= 0 , W R= 1, R E S E T =0 , C S =1
- (d) **R D= 0 , W R= 1, R E S E T =0 , C S =0**

103 . Which of these key switches is non mechanical contact type?

- (a) MECHANICAL KEY SWITCHES
- (b) **HALL EFFECT KEY SWITCHES**
- (c) MAGNETIC REED KEY SWITCHES
- (d) MEMBRANE KEY SWITCHES

104 . How many steps per revolution does a stepper motor in full step mode take?

- (a) 220 (b) **200** (c) 400 (d) 380

105 . An A/D converter which has a maximum output digital value of 1023 need show many wires for connecting its digital output to a peripheral chip

- (a) **10** (b) 13 (c) 12 (d) 8

106 . Which of these is used for Power brown out management?

- (a) Hold
- (b) **NMI**
- (c) Irq
- (d) Reset

107 . Which of these is called when a Divide Error occurs?

- (a) INT3
- (b) INT1
- (c) INT2
- (d) **INT0**

108 . DOS Interrupts reside in the?

- (a) ROM
- (b) Floppy Disk
- (c) Hard Disk
- (d) **RAM**

109 . For which one of these values of AL does the Video interrupt sets 80X25 color video mode?

- (a) 02H (b) 00H (c) **03H** (d) 01H

110 . Which of these Initialisation Command Word indicates whether slave connected to the PIC IRQ pins?

- (a) **ICW3**
- (b) ICW2
- (c) ICW4
- (d) ICW1

111. The device from which data originates or terminates is called

- (a) DSR (b) DTR (c) **DTE** (d) DCE

112. When the MODEM is ready to Transmit Data it Asserts which one of these signals is asserted?

- (a) DSR (b) CD (c) RTS (d) **CTS**

113. Given the program shown below, What is the expression evaluated by this program?

P db 5

Q db 6

R db 7

MOV AL,[P]

MUL AL

MOV BL,AL

MOV AL,[Q]

MUL AL

ADD AL,BL

MOV [R],AL

(a) **R=(P²+ Q²)**

(b) R=(P+Q)

(c) R= P+Q²

(d) R=(P+Q)²

114. The L2 and 1 bits in the Mode word are used for setting which of these parameters?

(a) NUMBER OF STOP BITS

(b) **CHARACTER LENGTH**

(c) BAUD RATE FACTOR

(d) PARITY

115. The data signals used by USB are of which type of the following?

(a) **Uni-Phase**

(b) Quad Phase

(c) Bi-Phase

(d) Tri-phase

116. The 8051 micro controller built in timer is of how many bits?

(a) **16** (b) 8 (c) 32 (d) 24

117. Which of these register of 8051 is 16 bit?

(a) **DPTR** (b) PSW (c) SP (d) A

118. One of these registers is loaded with the upper byte of the terminal count?

(a) TLx

(b) TCON

(c) **THx**

(d) TMOD

119. Which of the following is the correct order of priority in decreasing order from left to right?

(a) ET0,EX0,ET1,EX1,ES

(b) EX0,EX1,ET0,ET1,ES,

(c) **EX0,ET0,EX1,ET1,ES**

(d) ES,EX1,ET0,ET1,ET1

120. Which of these signals is used for enabling the external ROM?

(a) ALE

(b) RESET

(c) EA

(d) PSEN

121 . Which of these is Single Hand Shake

(a) STB followed by ACK and removal of ACK

(b) STB only

(c) **STB followed by ACK**

(d) ACK followed by STB

122 . At power on or after reset the ports are in which of these conditions

(a) **PORT A=INPUT,PORT B=INPUT,PORT C=INPUT**

(b) PORT A=INPUT,PORT B=OUTPUT,PORT C=OUTPUT

(c) PORT A=OUTPUT,PORT B=INPUT,PORT C=OUTPUT

(d) PORT A= INPUT,PORT B=INPUT, PORT C=OUTPUT

123 . Which of these key switches is non mechanical contact type ?

(a) MEMBRANE KEY SWITCHES

(b) **HALL EFFECT KEY SWITCHES**

(c) MECHANICAL KEY SWITCHES

(d) MAGNETIC REED KEY SWITCHES

124 . How many steps per revolution does a stepper motor in half step mode take?

(a) 200 (b) 220 (c) **400** (d) 380

125 . An 8 bit A/D converter with a reference voltage of 5 Volts will be able to read a lowest non zero voltage of

(a) 20 mV

(b) **19.5 mV**

(c) 19 mV

(d) 21 mV

126 . The third response of the 8086 to an interrupt is which one of the following?

(a) **Resets Trap Flag**

(b) Decrements the SP by 2&pushes flag register

(c) Decrements the SP & pushes current CS contents

(d) It disables interrupt flag

127 . What is the size of the vector interrupt table?

(a) 4096 bytes

(b) 256 bytes

(c) 65536 bytes

(d) **1024 bytes**

128 . In the single character display using Interrupt 21H,DL is the register which contains the character to display. If DL contains 61 H which of these characters is displayed?

(a) B (b) **a** (c) b (d) A

129 . For which one of these values of AH the key board interrupt wait for a key to be pressed before returning?

(a) 03H (b) 01H (c) 02H (d) **00H**

130. Which of these Initialisation Command Word indicates

whether PIC is being used in Single or Cascaded mode?

- (a) ICW3
- (b) ICW1**
- (c) ICW4
- (d) ICW2

131. The device from which data originates or terminates is called

- (a) DTE**
- (b) DCE
- (c) DSR
- (d) DTR

132. Serial data is received by the DTE from the DCE through which one of these signals?

- (a) RTS
- (b) CD
- (c) CTS
- (d) RxD**

133. Given the program shown below, What is the expression evaluated by this program?

```
P db 5
Q db 6
R db 7
MOV AL,[P]
MUL AL
MOV BL,AL
MOV AL,[Q]
MUL AL
ADD AL,BL
MOV [R],AL
```

- (a) $R=(P+Q)$
- (b) $R=(P+Q)^2$
- (c) $R=(P^2+Q^2)$**
- (d) $R=P+Q^2$

134. The L2 and 1 bits in the Mode word are used for setting which of these parameters?

- (a) CHARACTER LENGTH**
- (b) NUMBER OF STOP BITS
- (c) PARITY
- (d) BAUD RATE FACTOR

135. The data signals used by USB are of which type of the following?

- (a) Uni-Phase
- (b) Bi-Phase**
- (c) Quad Phase
- (d) Tri-phase

136. The size of the ROM in 8051 is

- (a) 4K bytes**
- (b) 16K bytes

(c) 32K bytes

(d) 8K bytes

137. Which of these registers cannot be decremented?

(a) Accumulator A

(b) DPTR

(c) B register

(d) SP

138. The lower byte of a 16 bit count is loaded in which of these registers?

(a) TLx

(b) THx

(c) TCON

(d) TMOD

139. The Serial Interrupt is enabled by which of these bits?

(a) ES (b) EA (c) ET (d) EX

140. This input pin is used for accessing Internal or External Memory and also generating the bus signals for micro processor operation of the micro controller?

(a) PSEN

(b) EA

(c) RESET

(d) ALE

141 . The 8255 PPI has

(a) 64 pins

(b) 16 pins

(c) 40 pins

(d) 28 pins

142 . If the control word 09BH is given to the control register of the 8255 PPI then which of these options are the condition of the ports

(a) BIT SET RESET

(b) ALL PORT's INPUT

(c) ALL PORT's OUTPUT

(d) PORT's A&B INPUT,PORT C OUTPUT

143 . If 8 seven segment led non multiplexed displays including the decimal point are connected. How many wires are needed to connect the segments(not including power supply wires)?

(a) 64 (b) 15 (c) 56 (d) 16

144 . What is the angle per step of a stepper motor in full step mode?

(a) 0.45 degrees

(b) 1.0 degrees

(c) 0.9 degrees

(d) 1.8 degrees

145 . In the A/D converter ADC 0808 which of these signals is used select an analog input channel to be digit is

(a) EOC (b) OE (c) **ALE** (d) SOC

146 . When an interrupt service is over, what is the instruction which makes 8086 continue executing the program it was executing before the Interrupt occurred

(a) JMP (b) ESC (**c**) **IRET** (d) RET

147 . What is the offset address of INT 20H in the vector interrupt table?

(a) 0050D

(b) 0050H

(c) 0080D

(**d**) **0080H**

148 . The DOS function AH=9 INT 21H displays a message till which one of this character is reached

(a) × (**b**) \$ (c) # (d) *

149 . Which one of these BIOS services is used for Printer Operation?

(a) 15H (b) 14H (**c**) **17H** (d) 16H

150. The PIC interrupt controller connects the single 8086 IRQ to how many IRQ input lines?

(a) 16 (**b**) **8** (c) 4 (d) 32

151. The UART used as Serial Communication controller is

(a) 8257 (b) 8251 (**c**) **8250** (d) 8259

152. How many registers does 8251 have?

(a) 3 (**b**) **2** (c) 1 (d) 4

153. Which one of these instructions could be used for swapping in a sort program

(**a**) **XCHG AX,DX**

(b) SBB AX,DX

(c) SUB AX_i DX

(d) MOV AX,BX

154. If B1=0 and B0=0 in the mode register of the 8251.The USART is working in which of these modes?

(a) Synchronous mode

(**b**) **Asynchronous mode with BRF=16**

(c) Asynchronous mode with BRF=64

(d) Asynchronous mode with BRF=1

155. The serial data transmission standards which uses differential Rx & Tx signals is which one of the following?

(a) RS232C

(b) RS449

(c) RS423A

(**d**) **RS422A**

156. The 8051 is a

(a) 16 bit micro controller

(b) 24 bit micro controller

(**c**) **8 bit micro controller**

(d) 32 bit micro controller

157. Which of the registers R0 to R7 support indirect

addressing?

- (a) R6 & R7
- (b) R0 & R6
- (c) R0 & R7
- (d) R0 & R1**

158. Which of these modes the timer is a 16 bit timer?

- (a) Mode 2
- (b) Mode 3
- (c) Mode 1**
- (d) Mode 0

159. The mode in which serial port operates like a shift register is which one of the following?

- (a) Mode0**
- (b) Mode2
- (c) Mode3
- (d) Mode1

160. The instruction used to access data from external memory is which one of the following?

- (a) MOV @
- (b) MOV
- (c) MOVC
- (d) MOV X**

161 . Group B port assignments of 8255 is

- (a) Port B & Lower Port C**
- (b) Port A & Upper Port C
- (c) Port B & Upper Port C
- (d) Port A & Lower Port C

162 . If the control word 00 is given to the control register of the 8255 PPI then which of these options are the condition of the ports

- (a) BIT SET RESET**
- (b) ALL PORT's OUTPUT
- (c) PORT's A&B INPUT, PORT C OUTPUT
- (d) ALL PORT's INPUT

163 . If 8 seven segment led non multiplexed displays including the decimal point are all lighted up and each of these is biased to carry 5mA.What is the total current drawn from the power supply when all the segments are ON?

- (a) 280 mA
- (b) 40 mA
- (c) 80 mA
- (d) 320 mA**

164 . The electro mechanical device which produces mechanical displacement in proportion to an applied voltage is know as

- (a) Solenoid
- (b) Contactor

(c) **Actuator**

(d) Relay

165 . In a D/A converter the output voltage is

(a) Voltage

(b) Pulses

(c) **current**

(d) Ramp

166 . When an interrupt service is over, what is the instruction which makes 8086 continue executing the program it was executing before the Interrupt occurred

(a) JMP

(b) **IRET**

(c) ESC

(d) RET

167 . Which of these is called Break Point Interrupt?

(a) INT2

(b) INT1

(c) INT0

(d) **INT3**

168 . Which of these values in AH performs output on video display of Ascii character in DL?

(a) 17 (b) 01 (c) 09 (d) **02**

169 . Which one of these BIOS services is used for Printer Operation?

(a) 16H (b) 14H (c) 15H (d) **17H**

170. The PIC interrupt controller which indicates that an Interrupt is already in service is?

(a) **ISR** (b) IRR (c) IMR (d) PR

171. Communications which take place in either direction between two systems, but can only occur in one direction at a time is known as

(a) Complex

(b) **Half Duplex**

(c) Full Duplex

(d) Simplex

172. The 8251 gets its clock from which of these signals?

(a) SYNDT/BD

(b) C/D

(c) **CLK**

(d) TxC

173. Which combination of the flags indicate that the source is equal to the destination

(a) CY=1,Z=1

(b) CY=0,Z=0

(c) CY=1,Z=0

(d) **CY=0,Z=1**

174. Which of these bits in status register is used to clear all errors?

(a) **ER** (b) EN (c) RTS (d) IR

175. The serial data transmission standards which uses differential Rx& Tx signals is which one of the following?

(a) RS449

(b) RS232C

(c) RS423A

(d) **RS422A**

176. The size of the internal data RAM in 8051 is

(a) **128 bytes**

(b) 256 bytes

(c) 32 bytes

(d) 1K bytes

177. Which of these register of 8051 is 16 bit?

(a) **DPTR**

(b) A

(c) PSW

(d) SP

178. Counter or timer operation is chosen in which of these registers?

(a) TLx

(b) THx

(c) TCON

(d) **TMOD**

179. Which of these modes is a standard UART mode?

(a) Mode3

(b) Mode0

(c) **Mode1**

(d) Mode2

180. The instruction used to access data from external memory is which one of the following?

(a) MOV @

(b) MOV

(c) **MOV X**

(d) MOVC

181 . Which of these is Double Hand Shake

(a) STB only

(b) STB followed by ACK

(c) **STB followed by ACK and removal of ACK**

(d) ACK followed by STB

182 . At power on or after reset the ports are in which of these conditions

(a) **PORT A=INPUT,PORT B=INPUT, PORT C=INPUT**

(b) PORT A=INPUT,PORT B=INPUT, PORT C=OUT PUT

(c) PORT A=INPUT, PORT B=OUTPUT, PORT C=OUTPUT

(d) PORT A=OUTPUT, PORT B=INPUT, PORT C=OUTPUT

183 . Which of these Peripheral ICs is made for interfacing Keyboards and Multiplexed LED displays?

(a) 8259 (b) 8237 (c) 8251 (d) **8279**

184 . Which of these motors works like a digital system

(a) DC MOTOR

(**b**) **STEPPER MOTOR**

(c) INDUCTION MOTOR

(d) SYNCHRONOUS MOTOR

185 . An 8 bit A/D converter with a reference voltage of 5 Volts will be able to read a lowest non zero voltage of

(a) 20 mV (**b**) **19.5 mV** (c) 21 mV (d) 19 mV

186 . The third response of the 8086 to an interrupt is which one of the following?

(a) Decrements the SP& pushes current CS contents

(**b**) **Resets Trap Flag**

(c) It disables interrupt flag

(d) Decrements the SP by 2& pushes flag register

187 . Which of these is called for doing Single Stepping?

(a) INT3

(**b**) **INT1**

(c) INT0

(d) INT2

188 . In the single character display using Interrupt 21H, DL is the register which contains the character to display. If DL contains 41H which of these characters is displayed?

(**a**) **A** (b) B (c) b (d) a

189 . For which one of these values of AL does the Video interrupt sets 80X25 color video mode?

(**a**) **03H** (b) 00H (c) 02H (d) 01H

190. Which of these Initialisation Command Word indicates whether PIC is being used in Single or Cascaded mode?

(a) ICW2

(b) ICW4

(**c**) **ICW1**

(d) ICW3

191. The device from which data originates or terminates is called

(a) DT R

(b) DCE

(**c**) **DTE**

(d) DSR

192. Serial data is received by the DTE from the DCE through which one of these signals?

(**a**) **RxD**

(b) CD

(c) RTS

(d) CTS

193. Given the program shown below, What is the expression evaluated by this program?

P db 5

Q db 6

R db 7
S db 6
MOV AL,[P]
MOV BL,[Q]
ADD AL,BL
MUL AL
ADD AL,R
MOV CL,S
DIV CL

- (a) $((P+Q)^2+R)/S$
- (b) $(P+Q^2+R)/s$
- (c) $((P +Q)^2+R)/S$**
- (d) $(P+(Q+R)^2)/S$

194. Which of these parameters are set by the S2& S1 bits?

- (a) BAUD RATE FACTOR
- (b) PARITY
- (c) CHARACTER LENGTH
- (d) NUMBER OF STOP BITS**

195. The data signals used by USB are of which type of the following?

- (a) Uni-Phase
- (b) Quad Phase
- (c) Bi - Phase**
- (d) Tri-phase

196. The 8051 is a

- (a) 24 bit micro controller
- (b) 16 bit micro controller
- (c) 8 bit micro controller**
- (d) 32 bit micro controller

197. Which of these registers cannot be decremented?

- (a) B register
- (b) SP
- (c) Accumulator A
- (d) DPTR**

198. Which of these modes the timer is a 16 bit timer?

- (a) Mode 3
- (b) Mode 0
- (c) Mode 2
- (d) Mode 1**

199. The mode in which serial port operates like a shift register is which one of the following?

- (a) Mode 0**
- (b) Mode 1
- (c) Mode 3
- (d) Mode 2

200. Which of these signals is used for enabling the external ROM ?

- (a) EA

- (b) RESET
- (c) ALE
- (d) PSEN

20. Tutorial classes

Hard copies will be attached

21. Known gaps

1. The MPMC subject as per the curriculum is not matching with the advance microcontrollers like AVR,PIC and ARM.
2. As per industry applications the known gap of TTL compatible devices like RS-232,GPIB is not present in MPMC which in the JNTU curriculum.
3. As per industry applications the known gap of DOS and BIOS Interrupts, 8259 Interrupt controller and DMA controller are not present in MPMC which is in JNTUH curriculum
4. As per industry applications the known gap of microprocessor real time applications are not present in MPMC subject which in the JNTU curriculum like home applications.

Actions taken:

The following topics are taken to fill the known gaps

- 1.RISC vs CISC ,AVR ,ARM,PIC architectures
2. RS-2332,& GPIB
3. DOS & BIOS INTERRUPTS,8279,8257
4. Real time applications.

22. Discussion topics

UNIT 1:

- 8086 architecture
- Flag register organization
- Pin diagram of 8086
- Minimum mode and maximum mode operations

UNIT-2:

- Instruction Formats

- Addressing Modes
- Instruction Set

UNIT-3

- 8255 Architecture
- A/D and D/A Interfacing
- Keyboard/Display Interfacing
- Memory Interfacing
- Interrupt structure of 8086
- Architecture of 8251 USART

UNIT-4:

- Microcontroller architecture
- Register organization of 8051
- Memory organization
- i/o ports
- memory interfacing

UNIT 5:

- Timer/counter operations
- Serial communication operation.

23. References, Journals, websites and E-links

Text books:-

1. Advanced Microprocessors and peripherals, A.K Ray and K.M Burch and TMH 2000
2. Micro controllers-Deshmukh, TMH
3. 8086 and 80286 microprocessors, hardware and software interfacing, -Walter A. Triebel Singh
4. Microprocessor architecture, programming and applications with 8085, by Ramesh Goankar
5. Advanced Microprocessors and Interfacing, Badri Ram, TMH

Reference Text Books:-

1. Microprocessor and interfacing-Douglas V.Hall, 2007
2. The 8088 and 8086 microprocessor-PHI, 4th Edition, 2003
3. Microcomputer systems, the 8086/8088 Family, architecture, Programming & Design, Yu-Chang Liu & Glenn A Gibson, PHI
4. The Intel Microprocessor, 8086/8088, 80186, 80286, 80386 and 80486 programming and interfacing by Barry B. Brey

5. Microprocessors, Theory and Applications, Intel and Motorola – Rafiquzzaman
6. Atmel AVR microcontroller primer: programming and interfacing- steven f. barret, Daniel j. pack

Websites:-

1. www.embedded-computing.com
2. www.mcjournal.com
3. www.atmel.com
4. www.keil.com
5. <http://8085projects.info/Interfacing-of-PIC-8259-with-8085.html>

Journals:-

1. www.mcjournal.com (web journal on microcontrollers)
2. Microprocessors and Microcomputer System
3. Embedded Hardware Design

24. Quality Control Sheets

Hard copy will be attached

25. Students List

3-1 A:

S.No	Roll No	StudentName
1	13R11A0401	ACHANTA RAJKUMAR
2	13R11A0402	AITHA RAVALI
3	13R11A0404	ANDEM NANDINI
4	13R11A0405	BARLA HARIKA PRASANNA
5	13R11A0406	BEJJALA MANASA LAKSHMI PRASANNA
6	13R11A0407	BHEEMREDDY VINEETH REDDY
7	13R11A0408	BANALA AJAY
8	13R11A0409	BELIDE SAI SHARAN
9	13R11A0410	BODIGA JEEVAN ADITYA GOUD

10	13R11A0411	CHILUKA MADHAN KUMAR
11	13R11A0412	D AISHWARYA
12	13R11A0413	DAKURI SHUSHMA
13	13R11A0414	DANABOINA VAISHALI
14	13R11A0415	DHIRAVATH NEERU
15	13R11A0416	ERLAPALLY VISHNUVARDHAN GOUD
16	13R11A0417	GUMEDELLY SAI SHARAN
17	13R11A0418	GANNU TEJA REDDY
18	13R11A0419	GODAVARTHI MANIKANTA
19	13R11A0420	GORRIPATI PRASANTH
20	13R11A0421	GUNDA MAMATHA
21	13R11A0422	KATGI RENUKA
22	13R11A0423	KOLA LALITH KUMAR
23	13R11A0424	KORNEPATI AISHWARYA
24	13R11A0425	KOTHA SHIRISHA
25	13R11A0426	LINGA DIVYAVANI
26	13R11A0427	M BRAHMA KARTHEEK
27	13R11A0428	MANTRALA VENKATESH
28	13R11A0429	SIKKEM MOKSH DIPIKA
29	13R11A0430	NARALA SAI TARUN
30	13R11A0431	PADUMATI MAHENDER REDDY
31	13R11A0432	P VENUGOPAL
32	13R11A0433	PALAPATI ADITHI
33	13R11A0434	PEDDINTI RATHAN RAJU
34	13R11A0435	PENDLI PRASHANTHI
35	13R11A0436	PERUMALLA PRASHANTH SAGAR
36	13R11A0437	POTTABATHINI DHARANI AISHWARYA
37	13R11A0438	RAMIDI MOUNIKA
38	13R11A0439	RAVALI UPADHYA
39	13R11A0440	RISHI BHARADWAJ BHUVANAGIRI
40	13R11A0441	ROHIT KUMAR REDDY K
41	13R11A0442	S VINAY SAI

42	13R11A0443	S. SREEJA SRINIVAS
43	13R11A0444	SUNCHU VISHWANADH NATARAJ
44	13R11A0445	SANAM RAJIT KUMAR
45	13R11A0446	SINGIREDDY SINDHUJA
46	13R11A0447	SINGI REDDY VEENITH KUMAR REDDY
47	13R11A0448	SUVIDHA K
48	13R11A0449	T VIJASH CHANDRA
49	13R11A0450	V. SAI CHARAN
50	13R11A0451	VAJROJU ARUN KUMAR
51	13R11A0452	MANDHA SANGEETHA
52	13R15A0409	BONAGIRI VEERESH

ECE-B

S.No	Roll No	StudentName
1	13R11A0453	A MADHURIMA
2	13R11A0454	ABDUL RAHMAN
3	13R11A0455	AKULA SWATHI
4	13R11A0456	ANANTHULA AKSHITHA
5	13R11A0457	BINGI SHEKAR
6	13R11A0458	BANDI NIHAL
7	13R11A0459	BATHINI SINDHURA
8	13R11A0460	BELIDE SURESH
9	13R11A0461	BHUKYA SANGU SAI KUMAR
10	13R11A0462	CH REVANTH REDDY
11	13R11A0463	CHANDANA CHAKRAVARTHY
12	13R11A0464	DASARI BHAVYA SRI
13	13R11A0465	GATTU VINEETH REDDY
14	13R11A0466	GANGAPURAM DILEEP KUMAR GOUD
15	13R11A0467	GUDE LAXMINARASIMHA

16	13R11A0468	GAINI S PRAVEEN KUMAR
17	13R11A0469	GUNJI RAVI DEVA RAJ
18	13R11A0470	GUNTUKA PRANAYA
19	13R11A0471	KANUKOLLU S V S RAMANA KIRAN
20	13R11A0472	KOUDDEVI SUDHAKAR
21	13R11A0473	KURUSALI SREE NIDHI
22	13R11A0474	LINGAMPALLI VIVEKAR
23	13R11A0475	MARTHA MOUNIKA
24	13R11A0476	M SRINIDHI
25	13R11A0477	MALLESWARAPU VINUSHA
26	13R11A0478	MONICA JADHAV
27	13R11A0479	MUTHALE ANIL
28	13R11A0480	N MAHITHA REDDY
29	13R11A0481	N YAMINI
30	13R11A0482	NIDAMANURI SAMEERA
31	13R11A0483	NIKHILA BADDAM
32	13R11A0484	PHATALE AKSHAY KUMAR
33	13R11A0485	P PRIYANKA
34	13R11A0486	PV GAYATHRI
35	13R11A0487	PATHI VENKATESH
36	13R11A0488	PONDUGULA BHARGAVI
37	13R11A0489	POTHIREDDY ANUSHA
38	13R11A0490	RENJU K R
39	13R11A0491	S ASHWITH REDDY
40	13R11A0492	SARITA KUMARI
41	13R11A0493	GAYATRI T
42	13R11A0494	T SUMA
43	13R11A0495	TALARI NEETHIKA
44	13R11A0496	V ANJANI KUMAR
45	13R11A0497	V S SRINATH
46	13R11A0498	VANGETI AMULYA REDDY
47	13R11A0499	VEERA ABHINAY GOUD
48	13R11A04A0	PAIDIPALA ANUSHA
49	13R11A04A1	RUDHRA RAJU SAI SUMANTH

50	14R15A0401	GUDE KALYAN KUMAR
51	14R15A0402	BANOTH RAJESH NAIK
52	14R15A0403	CHINDAM OMKAR
53	14R15A0404	ANNAM HARSHITH
54	14R15A0405	PAKALA CHAITANYA
55	14R15A0406	P SKANDHANA
56	14R15A0407	BANOTH ASHOK
57	14R15A0408	RAMAVATH LALU

ECE-C

S.No	Roll No	StudentName
1	13R11A04A3	AKHILA R KULKARNI
2	13R11A04A4	ANDE PREM BABU
3	13R11A04A5	AVULA SAI MEGHANA
4	13R11A04A6	B ADITHYA
5	13R11A04A7	BURRA NIKHIL GOUD
6	13R11A04A8	BUGGA RAGHU
7	13R11A04A9	BAJANTRI UMA DEVI
8	13R11A04B0	BAYYANI SAI SURYA
9	13R11A04B1	BETALA KONDAL RAO
10	13R11A04B2	BYREDDY SHIVANI
11	13R11A04B3	CHITTOOR RAVICHANDRAN KEERTHI KUMAR
12	13R11A04B4	G D HAREESH KUMAR
13	13R11A04B5	GINJUPALLI SAVITHRI MANI
14	13R11A04B6	GIREDDY KEERTHANA REDDY
15	13R11A04B7	GONURI SIRI CHANDANA
16	13R11A04B8	JASTI PANIDHAR
17	13R11A04B9	K MOUNICA
18	13R11A04C0	KALIDINDI GAYATHRI
19	13R11A04C1	KODIMALA SATISH REDDY

20	13R11A04C2	KANDHADI SHIVA REDDY
21	13R11A04C3	KULAKCHERLA RAGINI
22	13R11A04C4	KUNDUR NIKHIL
23	13R11A04C5	KUSHANGALA BHANU
24	13R11A04C6	MARYALA BALAKRISHNA
25	13R11A04C7	MANASA A
26	13R11A04C8	MANDA ANUSHA
27	13R11A04C9	MANDAVA ANIRUDH RANA
28	13R11A04D0	MANDULA NAGA RANI
29	13R11A04D1	MANEPALLI HARIKA
30	13R11A04D2	N SUJITH
31	13R11A04D3	N TANMAYEE
32	13R11A04D4	N ARJUN KUMAR
33	13R11A04D5	NAGARLA VIVEK VARDHAN
34	13R11A04D6	NEMOORI DINESH GOUD
35	13R11A04D7	NEMOORI LIKITH
36	13R11A04D8	PASTULA KARTIKEYA
37	13R11A04D9	PRIYANKA SOLANKI
38	13R11A04E0	PULIGOPU NIHARIKA
39	13R11A04E1	RAMINI PUJITHA
40	13R11A04E2	S SAI ABHINAV
41	13R11A04E3	S SRAVAN SETHURAM
42	13R11A04E4	SAMA MANOJ KUMAR
43	13R11A04E5	T RAVALI
44	13R11A04E6	TUMMALA SAI KRISHNA GOUD
45	13R11A04E7	TUMMALA SHIVA KUMAR
46	13R11A04E8	THONUKUNURI MANOJ KUMAR
47	13R11A04E9	VARKALA SAI KIRAN
48	13R11A04F0	VAISHNAVI SRIDHAR
49	13R11A04F1	PANTHULU SINDHUJA
50	13R11A04F2	RAVI AVINASH
51	13R11A04F3	UPPLAPATI ANUSHA
52	13R11A04F4	SURAJ KANDURI

ECE-D

S.No	Roll No	StudentName
1	13R11A04F5	A RAMA THEJA
2	13R11A04F7	ANUGU PRASHANTH
3	13R11A04F8	ARACHANA DASH
4	13R11A04F9	CHAVALI NAGARJUNA
5	13R11A04G0	CHIGURUPATI MEENAKSHI
6	13R11A04G1	D SRI RAMYA
7	13R11A04G2	DEEKONDA RAJSHREE
8	13R11A04G3	G MANIDEEP
9	13R11A04G4	GATADI VADDE PREM SAGAR
10	13R11A04G5	GOGU JEEVITHA SPANDANA REDDY
11	13R11A04G6	GOLLURI SINDHUJA
12	13R11A04G7	GOPANABOENA SAI KRANTHI KUMAR
13	13R11A04G8	GUNTIMADUGU SAI RESHMA
14	13R11A04G9	K DARSHAN
15	13R11A04H0	K. ANIRUDH
16	13R11A04H1	KOMIRISHETTY AKHILA
17	13R11A04H2	KOPPU MOUNIKA
18	13R11A04H3	KANNE RAVI KUMAR
19	13R11A04H4	KARRA VINEELA
20	13R11A04H5	KANUKALA SIDDHARTH
21	13R11A04H6	KATHI SHIVARAM REDDY
22	13R11A04H7	KOMANDLA SRIKANTH REDDY
23	13R11A04H8	KONDAM PADMA
24	13R11A04H9	KRISHNA ASHOK MORE
25	13R11A04J0	LINGAMPALLY RAJASRI
26	13R11A04J1	M ROHITH SAI SHASHANK
27	13R11A04J2	M TANVIKA
28	13R11A04J3	MALIHA AZAM
29	13R11A04J4	MANSHA NEYAZ

30	13R11A04J5	MATTA SRI SATYA SAI GAYATHRI
31	13R11A04J6	MD RAHMAN SHAREEF
32	13R11A04J7	MEESALA SAI SHRUTHI
33	13R11A04J8	P G CHANDANA
34	13R11A04J9	PALLE AKILA
35	13R11A04K0	PERNAPATI YAMINI
36	13R11A04K1	POLISETTY VEDA SRI
37	13R11A04K2	REGU PRAVALIKA
38	13R11A04K3	R RITHWIK REDDY
39	13R11A04K4	RAMYA S
40	13R11A04K5	SURI BHASKER SRI HARSHA
41	13R11A04K6	TANGUTOORI SIRI CHANDANA
42	13R11A04K7	THIPPARAPU AKHIL
43	13R11A04K8	UDDALA DEVAMMA
44	13R11A04K9	VALASA SHIVANI
45	13R11A04L0	VEPURI NAGA TARUN SAI
46	13R11A04L1	VISWAJITH GOVINDA RAJAN
47	13R11A04L2	YENDURI YUGANDHAR
48	13R11A04L3	M SAI KUMAR
49	14R18A0401	MODUMUDI HARSHITHA

26 Group-Wise students list for discussion topics

Groups are formed after beginning of classwork