JEPPIAAR ENGINEERING COLLEGE

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

VISION OF INSTITUTION

To build Jeppiaar Engineering College as an institution of academic excellence in technology and management education, leading to become a world class University.

MISSION OF INSTITUTION

- To excel in teaching and **learning, research and innovation** by promoting the principles of scientific analysis and creative thinking.
- To participate in the production, **development, dissemination of knowledge** and interact with **national and international communities.**
- To equip students with ethical **values, and life skills** that would enrich their lives and enable them to meaningfully contribute to the **progress of the society.**
- To prepare students for **higher studies and lifelong learning,** enrich them with the **practical and entrepreneurial skills** necessary to excel as future professionals and contribute to **Nation's economy.**

PROGRAM OUTCOMES (POs)

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10 **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11 **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12 **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

VISION OF THE DEPARTMENT

The Department of Electrical and Electronics Engineering strives to be a Centre of Excellence in education and technical research, in the endeavour of which the Department will continually update the teaching methodologies, progress in the emerging technologies and continue to play a vital role in the development of the society.

MISSION OF THE DEPARTMENT

M1	To develop the ability to **learn** and work creatively that would enhance the ability of both students and faculty to do **innovative research**.
M2	To **create** and maintain state-of-the art facilities which provide students and faculty with opportunities to analyse, apply and **disseminate knowledge globally**.
M3	To impart the knowledge in essential interdisciplinary fields which will enhance the **interpersonal skills**, team work, professional **ethics** and make them work effectively for their own benefit and the **betterment of the society**.
M4	Prepare students for **lifelong learning** of theoretical and **practical concepts** to face intellectual, **economical** and career challenges.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO 01	Strengthen the knowledge in Electrical and Electronics Engineering to enable them **work** for modern industries by promoting energy conservation and sustainability.
PEO 02	Enrich analytical, creative and critical **logical reasoning** skills to solve problems faced by emerging domains of electrical and electronics engineering industries worldwide.
PEO 03	Develop effective communication and inter-personal skills to work with enhanced team spirit in multidisciplinary projects with a broader **ethical**, professional, economical and **social** perspective.
PEO 04	Prepare the students either to establish **start ups** or to pursue **higher education** at reputed institutions.

PROGRAM SPECIFIC OUTCOME (PSOs)

PSO 1	**Professional Skills:** Apply the knowledge of Mathematics, Science and Engineering to solve real time problems in the field of **Power Electronics, Electrical Drives, Power Systems, Control Systems and Instrumentation**.
PSO 2	**Research and Innovation:** Analyze and synthesize circuits by solving complex engineering problems to obtain the **optimal solution** using effective software tools and hardware prototypes in the field of robotics and renewable energy systems.
PSO 3	**Product development:** Develop concepts and **products** by applying ideas of electrical domain into other diversified engineering domains.

EE6008 MICROCONTROLLER BASED SYSTEM DESIGN L T P C

3 0 0 3

OBJECTIVES:

To introduce the architecture of PIC microcontroller

To educate on use of interrupts and timers

To educate on the peripheral devices for data communication and transfer

To introduce the functional blocks of ARM processor

To educate on the architecture of ARM processors

UNIT I INTRODUCTION TO PIC MICROCONTROLLER 9

Introduction to PIC Microcontroller–PIC 16C6x and PIC16C7x Architecture–PIC16cxx–- Pipelining - Program Memory considerations – Register File Structure - Instruction Set - Addressing modes – Simple Operations.

UNIT II INTERRUPTS AND TIMER 9

PIC micro controller Interrupts- External Interrupts-Interrupt Programming–Loop time subroutine - Timers-Timer Programming– Front panel I/O-Soft Keys– State machines and key switches– Display of Constant and Variable strings.

UNIT III PERIPHERALS AND INTERFACING 9

I^2C Bus for Peripherals Chip Access– Bus operation-Bus subroutines– Serial EEPROM— Analog to Digital Converter–UART-Baud rate selection–Data handling circuit–Initialization - LCD and keyboard Interfacing -ADC, DAC, and Sensor Interfacing.

UNIT IV INTRODUCTION TO ARM PROCESSOR 9

ARM Architecture –ARM programmer's model –ARM Development tools- Memory Hierarchy –ARM Assembly Language Programming–Simple Examples–Architectural Support for Operating systems.

UNIT V ARM ORGANIZATION 9

3-Stage Pipeline ARM Organization– 5-Stage Pipeline ARM Organization–ARM Instruction Execution- ARM Implementation– ARM Instruction Set– ARM coprocessor interface–Architectural support for High Level Languages – Embedded ARM Applications.

Total:45 Periods

OUTCOMES:

To understand and apply computing platform and software for engineering problems.

To understand ethical issues, environmental impact and acquire management skills.

TEXT BOOKS:

1. Peatman,J.B., "Design with PIC Micro Controllers"PearsonEducation,3rdEdition, 2004.
2. Furber,S., "ARM System on Chip Architecture" Addison Wesley trade Computer Publication, 2000.

REFERENCES:

1. Mazidi, M.A.,"PIC Microcontroller" Rollin Mckinlay, Danny causey Printice Hall of India, 2007.

Course code& Name: **EE6008 Microconytroller Based System Design**

Degree/Programme: **B.E/EEE** Semester: **VII** Section: **A, B**

Duration: **DEC – APRIL 2018** Regulation: **2013/AUC**

Name of the Staff:

Aim:

To understand and apply computing platform and software for engineering problems.

To understand ethical issues, environmental impact and acquire management skills.

OBJECTIVES:

To introduce the architecture of PIC microcontroller

To educate on use of interrupts and timers

To educate on the peripheral devices for data communication and transfer

To introduce the functional blocks of ARM processor

To educate on the architecture of ARM processors

C4 6.1	Understand the computing platform and software for engineering problem using PIC Microcontroller
C4 6.2	Explain about interrupts and timers of PIC Microcontroller
C4 6.3	Understand and utilize the peripheral devices for real time applications.
C4 6.4	Understand the basic structure and operation of ARM processor
C4 6.5	Explain about programming and applications of ARM processor.

EE6008	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO 12	PSO1	PSO2	PSO3
C4 6.1	2	3	2	1	-	3	1	-	-	-	-	3	1	3	3
C4 6.2	2	-	2	3	2	-	1	-	-	-	-	3	1	3	2
C4 6.3	3	3	-	3	3	-	1	-	-	-	-	3	1	3	2
C4 6.4	3	-	2	3	3	2	1	-	-	-	-	3	1	3	2
C4 6.5	2	-	2	3	3	2	1	-	-	-	-	3	1	3	3

UNIT-I	INTRODUCTION TO PIC MICROCONTROLLER	Target Period :9				
Sl N o	Contents	CO Statement	Book Reference & Page No	Delivery method	Delivery Periods	Knowledge Level
1	Introduction to PIC Microcontroller,PIC 16C6x and PIC16C7x	C4 6.1	T1[2-7]	Chalk & board / PPT	2	R & U
2	Pipelining	C4 6.1	T1[13-14]	Chalk & board / PPT	1	R & U
3	Program Memory considerations	C4 6.1	T1[14-18]	Chalk & board / PPT	1	R, U
4	Register File Structure & Addressing modes	C4 6.1	T1[18-21]	Chalk & board / PPT	1	R, U
5	Instruction Set	C4 6.1	T1[24-28]	Chalk & board / PPT	2	R, U
6	Simple Operations.	C4 6.1	T1[28-29]	Chalk & board / PPT	2	R,U,A
UNIT II	INTERRUPTS AND TIMER	Target Periods: 9				
Sl No	Contents	CO Statement	Book Reference & Page No	Delivery method	Delivery Hrs	Knowledge Level

1	PIC micro controller Interrupts micro controller Interrupts	C4 6.2	T1[58]	Chalk & board / PPT	**2**	R, U
2	External Interrupts	C4 6.2	T1[95-100]	Chalk & board / PPT	**1**	R, U
3	Interrupt Programming-Loop time subroutine	C4 6.2	T1[60-67]	Chalk & board / PPT	**2**	R, U, A
4	Timers-Timer Programming	C4 6.2	T1[100-120]	Chalk & board / PPT	**1**	R, U, A
5	Front panel I/O-Soft Keys	C4 6.2	T1[143-145]	Chalk & board / PPT	**1**	R, U
6	State machines and key switches	C4 6.2	T1[145-148]	Chalk & board / PPT	**1**	R,U
7	Display of Constant and Variable strings.	C4 6.2	T1[149-156]	Chalk & board / PPT	**1**	R, U

UNIT III PERIPHERALS AND INTERFACING Target Periods: 9

Sl No	**Contents**	**CO Statement**	**Book Reference & Page No**	**Delivery method**	**Delivery Hrs**	**Knowledge Level**
1	I2C Bus for Peripherals Chip Access	C4 6.3	T1[163-165]	Chalk & board / PPT	**1**	R, U
2	Bus operation-Bus subroutines	C4 6.3	T1[166-170]	Chalk & board / PPT	**1**	R, U
3	Serial EEPROM	C4 6.3	T1[174-178]	Chalk & board / PPT	**1**	R, U
4	Analog to Digital Converter	C4 6.3	T1[181-189]	Chalk & board / PPT	**1**	R, U
5	UART-Baud rate selection	C4 6.3	T1[190-195]	Chalk & board / PPT	**1**	R, U,
6	Data handling circuit-Initialization	C4 6.3	T1[196-200]	Chalk & board / PPT	**1**	R, U
7	LCD and keyboard Interfacing	C4 6.3	R1[473-495]	Chalk & board / PPT	**1**	R, U
8	ADC, DAC, and Sensor Interfacing.	C4 6.3	R1[499-524]	Chalk & board / PPT	**2**	R, U

UNIT IV INTRODUCTION TO ARM PROCESSOR Target Periods:9

Sl No	**Contents**	**CO Statement**	**Book Reference & Page No**	**Delivery method**	**Delivery Hrs**	**Knowledge Level**
1	ARM Architecture	C4 6.4	T2[36-38]	Chalk & board / PPT	**1**	R, U
2	ARM programmer’s model	C4 6.4	T2 [39-43]	Chalk & board / PPT	**1**	R, U
3	ARM Development tools	C4 6.4	T2 [43-46]	Chalk & board / PPT	**2**	R, U
4	Memory Hierarchy	C4 6.4	T2 [270-288]	Chalk & board / PPT	**1**	R, U
5	ARM Assembly Language Programming	C4 6.4	T2 [50-68]	Chalk & board / PPT	**1**	R, U

6	Simple Examples	C4 6.4	T2 [68-72]	Chalk & board / PPT	**2**	R, U, A
7	Architectural Support for Operating systems	C4 6.4	T2 [291-315]	Chalk & board / PPT	**1**	R, U
UNIT V	**ARM ORGANIZATION**			**Target Periods: 9**		
Sl No	**Contents**	**CO Statement**	**Book Reference & Page No**	**Delivery method**	**Delivery Hrs**	**Knowledge Level**
1	3-Stage Pipeline ARM Organization	C4 6.5	T2[75-78]	Chalk & board / PPT	**1**	R, U,
2	5-Stage Pipeline ARM Organization	C4 6.5	T2[78-82]	Chalk & board / PPT	**1**	R, U
3	ARM Instruction Execution	C4 6.5	T2[82-85]	Chalk & board / PPT	**1**	R, U
4	ARM Implementation	C4 6.5	T2[86-103]	Chalk & board / PPT	**1**	R, U
5	ARM Instruction Set	C4 6.5	T2[106-150]	Chalk & board / PPT	**1**	R, U
6	ARM coprocessor interface	C4 6.5	T2[152-184]	Chalk & board / PPT	**1**	R, U
7	Architectural support for High Level Languages	C4 6.5	T2[152-184]	Chalk & board / PPT	**2**	R, U
8	Embedded ARM Applications	C4 6.5	T2[348-360]	Chalk & board / PPT	**1**	R, U, A

Books: Text (T) / Reference(R):

S.No	Book No	**Title of the Book**	**Author**	**Publisher**	**Year**
1	T1	Design with PIC Micro Controllers"PearsonEducation	Peatman,J.B.,	PearsonEducation,3rd Edition	2004
2	T2	ARM System on Chip Architecture	Furber,S.	AddisonWesley trade Computer Publicatio	2000
3	R1	PIC Microcontroller	Mazidi.M	Rollin Mckinlay, Danny causey Printice Hall of India	2007

Comments Given by the Scrutinizing Committee Members	
Signature of the Scrutinizing	
Signature of the HOD	

EE6008-Microcontroller Based System Design

UNIT – I Introduction to PIC Microcontroller

PART – A

1. What is a Microcontroller?

A device which contains the microprocessor with integrated peripherals like memory, serial ports, parallel ports, timer/counter, interrupt controller, data acquisition interfaces like ADC, DAC is called a microcontroller.

2. What are the differences between a Microcontroller and Microprocessor?

S.No	Microprocessor	Microcontroller
1	It is termed as general purpose digital computer.	It is termed as special purpose digital controller.
2	It contains the CPU, memory, addressing circuits and interrupt handling circuit.	It possesses all features of microprocessor and additionally it includes timers, parallel and serial I/O and the internal RAM and ROM.
3	It has one or two types of bit handling instructions.	It has many bit handling instructions.

3. What is PIC Microcontroller?

PIC stands for Peripheral Interface Controller given by Microchip Technology to identify its single-chip microcontrollers. These devices have been very successful in 8-bit microcontrollers. The main reason is that Microchip Technology has continuously upgraded the device architecture and added needed peripherals to the microcontroller to suit customers' requirements.

4. Difference between Microcontroller and PIC microcontroller.(Nov/Dec-2017)

S.No	Microcontroller	PIC microcontroller
1.	It possesses all features of microprocessor and additionally it includes timers, parallel and serial I/O and the internal RAM and ROM & it doesn’t have inbuit A/D converter	PIC Microcontroller is an integrated chip which is consists of RAM, ROM, CPU, TIMER and COUNTERS & A/D converter.
2.	It supports for CSIC architecture	It used modified Harvard architecture and also supports RISC.

5. List out the factors controlling the popularity of PIC Microcontroller.

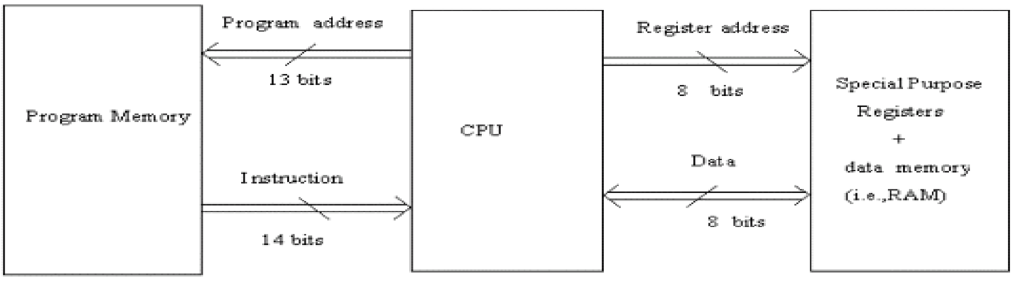
1. Speed: Harvard Architecture, RISC architecture, 1 instruction cycle = 4 clock cycles.
2. Instruction set simplicity: The instruction set consists of just 35 instructions (as opposed to 111 instructions for 8051).
3. Power-on-reset and brown-out reset. Brown-out-reset means when the power supply goes below a specified voltage (say 4V), it causes PIC to reset; hence malfunction is avoided.

   A watch dog timer (user programmable) resets the processor if the software/program ever malfunctions and deviates from its normal operation.
4. PIC microcontroller has four optional clock sources.
   a. Low power crystal
   b. Mid range crystal
   c. High range crystal
   d. RC oscillator (low cost).

5. Programmable timers and on-chip ADC.
6. Up to 12 independent interrupt sources.
7. Powerful output pin control (25 mA (max.) current sourcing capability per pin.)
8. EPROM/OTP/ROM/Flash memory option.
9. I/O port expansion capability.
10. Free assembler and simulator support from Microchip.

6. Draw the CPU architecture of PIC Microcontroller.

The CPU uses Harvard architecture with separate Program and Variable (data) memory interface. This facilitates instruction fetch and the operation on data/accessing of variables simultaneously.



CPU Architecture of PIC microcontroller

7. Define the term register file in PIC terminology.

The term register file is used to denote the locations that an instruction can access via an address. The register file contains two components:

1. General purpose register file
2. Special purpose register file.

8. **List out the types of addressing modes? (Nov/Dec-2017)**

1. Direct addressing mode
2. Indirect addressing mode

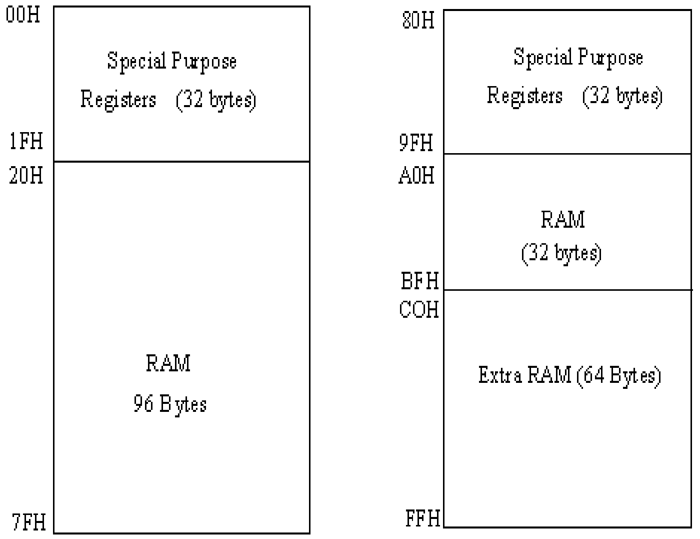
9. What is General purpose register file?

The General purpose register file is another name for the microcontrollers RAM. Data can be written to each 8 bit location, updated and retrieved any number of times.

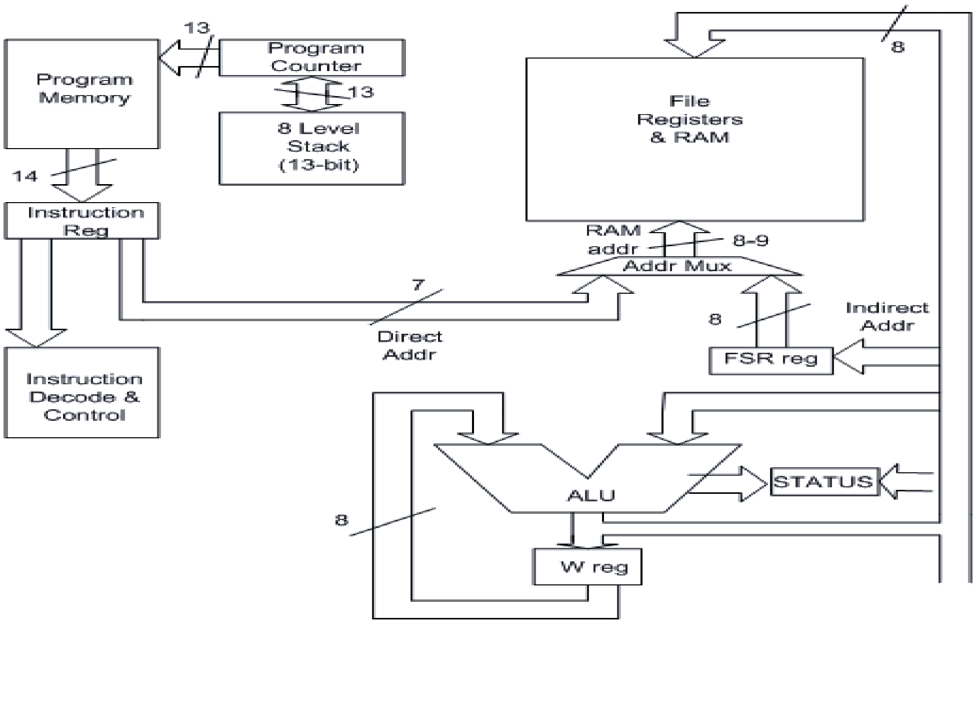
10. What is Special Purpose register file?

The special purpose register file contains input and output ports as well as the control registers used to establish each bit of a port as either an input or an output. It contains registers that provide the data input and data output to the variety of resources on the chip, such as the timers, the serial ports and the ADC. It has registers that contain control bits for selecting the mode of operation of a chip resource as well as enabling or disabling its operation. It has registers containing status bits, which denote the state of one of these chip resources.

11. Give the register file structure of PIC Microcontroller.



12. Draw the architecture of PIC 16C74A



13. What is 'W' register in PIC Microcontroller?

W, the working register, is used by many instructions as the source of an operand. This is similar to accumulator in 8051. It may also serve as the destination for the result of the instruction execution. It is an 8 - bit register.

W, Working register

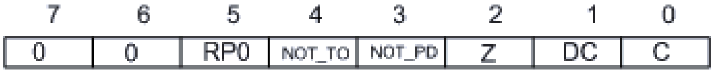
14. What are the benefits of having RISC architecture? (April/May-2017)

- Simplified Instruction Set
- 2 to 4 times more performance than CSIC instruction set.
- Less chip space required
- Fast time to design

15. Give the status register of PIC Microcontroller. (Nov/Dec 2016)

The STATUS register is an 8-bit register that stores the status of the processor. This also stores carry, zero and digit carry bits.

STATUS - address 03H, 83H



7	6	5	4	3	2	1	0
0	0	RP0	NOT_TO	NOT_PD	Z	DC	C

16. Explain FSR and INDF register.

FSR Register(File Selection Register, address = 04H, 84H)

FSR is an 8-bit register used as data memory address pointer. This is used in indirect addressing mode.

INDF Register(Indirect through FSR, address = 00H, 80H)

INDF is not a physical register. Accessing INDF access is the location pointed to by FSR in indirect addressing mode.

17. Explain PCL and PCLATH Register.

PCL Register

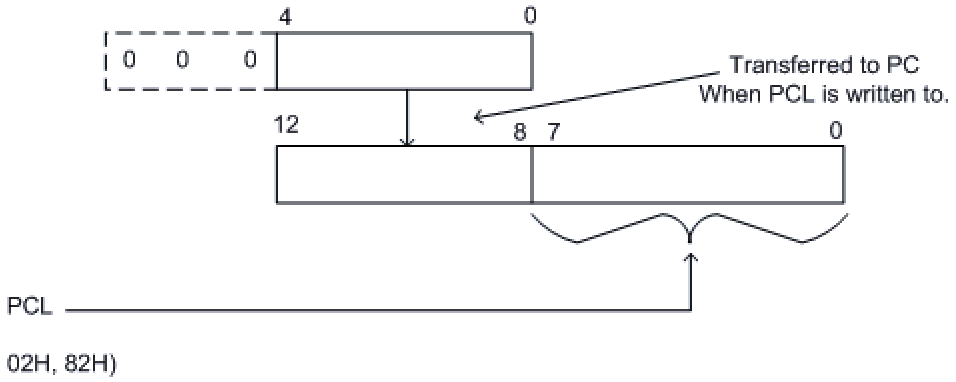
(Program Counter Low Byte, address = 02H, 82H)

PCL is actually the lower 8-bits of the 13-bit program counter. This is a both readable and writable register.

PCLATH Register

(Program Counter Latch, address = 0AH, 8AH)

PCLATH is a 8-bit register which can be used to decide the upper 5bits of the program counter. PCLATH is not the upper 5bits of the program counter. PCLATH can be read from or written to without affecting the program counter. The upper 3bits of PCLATH remain zero and they serve no purpose. When PCL is written to, the lower 5bits of PCLATH are automatically loaded to the upper 5bits of the program counter, as shown in the figure.



18. What is program counter stack?

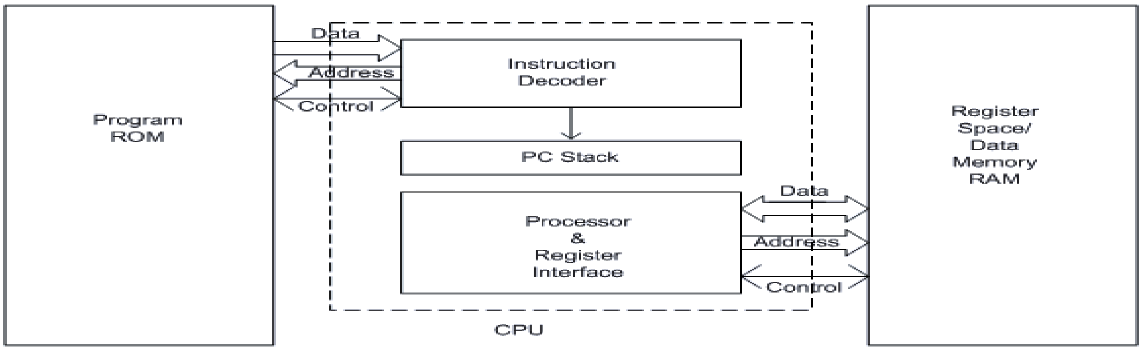
Program Counter Stack

An independent 8-level stack is used for the program counter. As the program counter is 13bit, the stack is organized as 8x13bit registers. When an interrupt occurs, the program counter is pushed onto the stack. When the interrupt is being serviced, other interrupts remain disabled. Hence, other 7 registers of the stack can be used for subroutine calls within an interrupt service routine or within the mainline program.

19. Define Brown out Reset

A drop in the voltage f electrical power supply is called Brownout. If V_{DD} falls below 4V for greater than 100ns the brownout situation is said to have occurred.This situation causes reset of the PIC16C62A.A brown-out reset will not occur if V_{DD} falls below 4V for less than 100ns.The typical value of BV_{DD} (Brown-out Reset voltage) is 4V

20. Draw the general block diagram of Harvard architecture.



21. Using the instruction of PIC microcontroller, convert BCD to Hex.

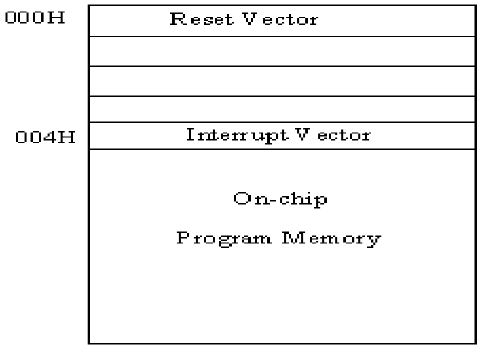
```
MOVLW 0X54
ADDLW 0X87
DAA.
```

22. How is the internal RAM in PIC microcontroller accessed by indirect addressing?

```
LFSR      0,0x30
LFSR 1, 0x40
```

23. Explain the PIC memory organization.

PIC microcontroller has 13 bits of program memory address. Hence it can address up to 8k of program memory. The program counter is 13-bit. PIC 16C6X or 16C7X program memory is 2k or 4k. While addressing 2k of program memory, only 11- bits are required. Hence two most significant bits of the program counter are ignored. Similarly, while addressing 4k of memory, 12 bits are required. Hence the MSB of the program counter is ignored.



24. What are the types of instruction set used in PIC microcontroller?

There are three types of instruction set used in PIC microcontroller.

1. Bit oriented instruction
2. 2. Byte oriented instruction
3. 3. Literal instructions.

25. What is bit and byte oriented instruction?

The **byte oriented instructions** that require two parameters (For example, movf f, F(W)) expect the f to be replaced by the name of a special purpose register (e.g., PORTA) or the name of a RAM variable (e.g., NUM1), which serves as the source of the operand. 'f' stands for file register. The F(W) parameter is the destination of the result of the operation. It should be replaced by:F, if the destination is to be the source register.

W, if the destination is to be the working register (i.e., Accumulator or W register).

The **bit oriented instructions** also expect parameters (e.g., btfsc f, b). Here 'f' is to be replaced by the name of a special purpose register or the name of a RAM variable. The 'b' parameter is to be replaced by a bit number ranging from 0 to 7.

For example:

```
Z equ 2
btfsc STATUS, Z
```

Z has been equated to 2. Here, the instruction will test the Z bit of the STATUS register and will skip the next instruction if Z bit is clear.

26. Define literal instructions in PIC microcontroller.

The **literal instructions** require an operand having a known value (e.g., 0AH) or a label that represents a known value.

For example:

```
NUM equ 0AH ;    Assigns 0AH to the label NUM ( a constant )
movlw NUM ;      will move 0AH to the W register.
```

Every instruction fits in a single 14-bit word. In addition, every instruction also executes in a single cycle, unless it changes the content of the Program Counter. These features are due to the fact that PIC micro controller has been designed on the principles of RISC (Reduced Instruction Set Computer) architecture.

27. What is I/O port of PIC?

I/O port is used to get and send the data from/to external devices. Some I/O pins have multifunction.

28. List the significance of brown out reset mode?

When the power supply falls below a certain voltage, it causes PIC to reset. This is called as brown out to reset mode.

29. What are the addressing modes of PIC? (Nov/Dec 2016)

Addressing is defined as how the operands are specified in the instruction. Direct addressing and indirect addressing mode.

30. What do you mean by direct addressing mode?

It uses 7 bits of the instruction and the 8th bit from RP. It directly give the address where the data is present.ie, the address of the operand is given in the instruction.

31. What is instruction pipelining?

It allows the CPU to fetch and execute at the same time while executing one instruction, CPU will fetch next instruction to be executed.

32. What is indirect addressing mode?

Indirectly addressing the memory used in FSR and INDF instruction. Here the operand is specified indirectly in the instruction.

33. What is Access bank in PIC18?

It is the default bank which is invoked when power up. It is divided into two equal sections of 128 bytes, which is given to GPR and SFR

34. Give the importance of MCLR?

It is called master clear pin, this pin is used to reset the PIC.

35. Write any four instructions of PIC microcontroller and state in a line the Operation performed.

```
MOVLW 25H        ;     WREG=25
ADDLW 0X34       ;     ADD 34H TO WREG
ADDLW            ;     ADD 11H TO WREG
ADDLW            ;     W=W+12H=7CH
```

PART-B

1. Explain with neat diagram the architecture of PIC16C6x and PIC16C7x microcontroller. **(Nov/Dec 2016) , (Nov/Dec-2017), (April/May-2017)**
2. Explain with neat diagram the block diagram of PIC16C6x and PIC16C7x microcontroller
3. With a neat diagram discuss in detail about memory organization of a PIC microcontroller. **(April/May-2017)**
4. Explain in detail the register file structure and addressing modes of PIC microcontroller?**(Nov/Dec-2017), (April/May-2017)**
5. Explain the instruction set of PIC microcontroller. **(Nov/Dec 2016)& (Nov/Dec-2017)**
6. Explain in detail the CPU register used in PIC microcontroller.
7. Write an ALP to divide two 8 bit numbers.
8. Write an ALP for arranging the sequence of numbers in ascending and descending order.
9. Explain in detail the concept of pipelining of instructions in PIC microcontroller.
10. Write an ALP to add and subtract two numbers in PIC.

UNIT- II INTERRUPTS AND TIMERS

PART-A

1. What are hardware and software interrupts?

PIC Microcontroller consists of both Hardware and Software Interrupts. If the interrupts are generated by external hardware at certain pins of microcontroller, or by inbuilt devices like timer, they are called Hardware Interrupts. While Software interrupts are generated by a piece of code in the program. Also known as external and internal interrupts.

2. What are the interrupts available in PIC? (Jan’14)& (April/May-2017)

Interrupt Source	Enabled by	Completion Status
External interrupt from	INT INTE = 1	INTF = 1
TMR0 interrupt	T0IE = 1	T0IF = 1
RB4–RB7 state change	RBIE = 1	RBIF = 1
EEPROM write complete	EEIE = 1	

3. How many timers are there in PIC?

Timers- timer0, 1 and 2.

4. **Brief timer0.**

It is an 8 bit wide timer internal clock is $f_{osc}/4$ and external clock is given at RA4 pin.

5. **How do you calculate the timer0 delay?**

Timer0 count x prescaler value x $4/f_{osc}$.

6. **How do you calculate timer0 preload count?**

256-(timer0delay x fosc)/prescalervalue x 4)

7. **What are the modes of timers?**

Rate generation waveform, square waveform are the modes involved.

8. **What are the features of timer0?**

The Timer0 module timer/counter has the following features:

- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable prescaler
- Internal (4 Mhz) or external clock select
- Interrupt on overflow from FFh to 00h
- Edge select (rising or falling) for external clock

9. **What are the features of timer 1?**

- The Timer1 module, timer/counter, has the following features:
- 16-bit timer/counter consisting of two 8-bit registers (TMR1H and TMR1L)
- readable and writable
- 8-bit software programmable prescaler
- Internal (4 Mhz) or external clock select
- Interrupt on overflow from FFFFh to 0000h

10. **What are the features of timer 2?**

- The Timer2 module, timer/counter, has the following features:
- two 8-bit registers (TMR2 and PR2)
- readable and writable
- a prescaler and a postscaler
- Connected only to an internal clock - 4 MHz crystal
- Interrupt on overflow

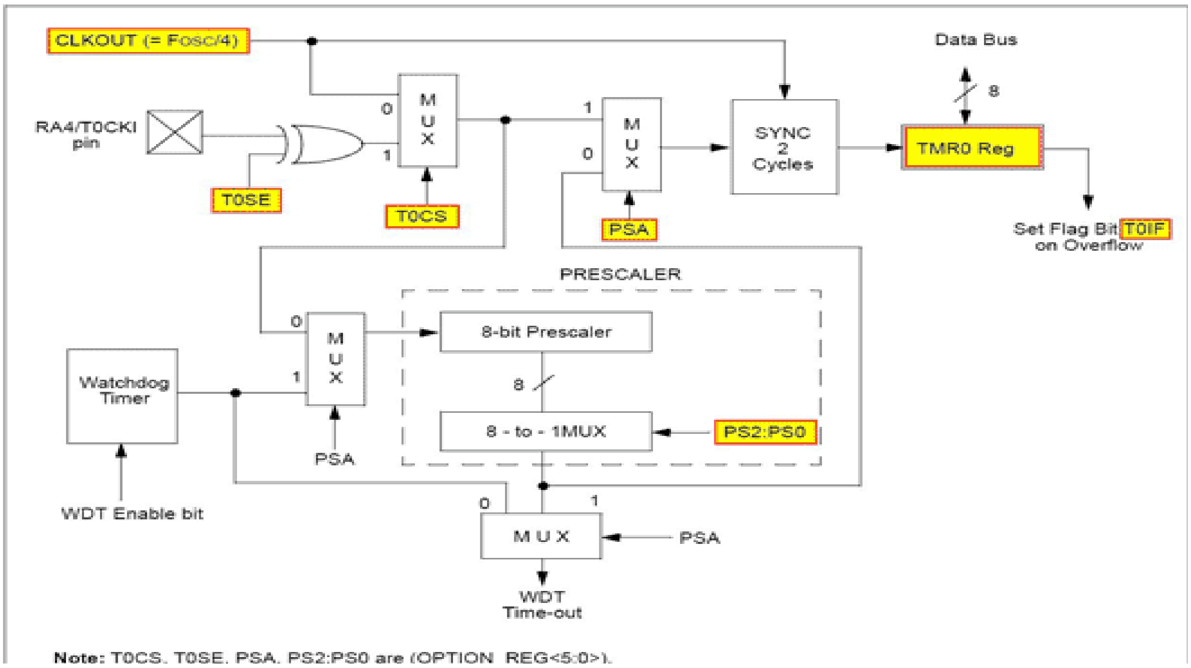
11. **What is TMR0 register?**

Timer0 has a register called TMR0 Register, which is 8 bits of size.The desired value can be written into the register which will be increment as the program progresses. Frequency varies depending on the Prescaler. Maximum value that can be assigned to this register is 255.

12. **How to initialize the option_reg register?**

PSA=0; // Prescaler is assigned to the Timer0 module
PS0=1; // Prescaler rate bits
PS1=1; // are set to “111”
PS2=1; // which means divide by 256
TOSE=0; // rising edge
TOCS=0; // Internal instruction cycle clock

13. **Draw the block diagram of PIC timer 0.**



Note: T0CS, T0SE, PSA, PS2:PS0 are (OPTION_REG<5:0>).

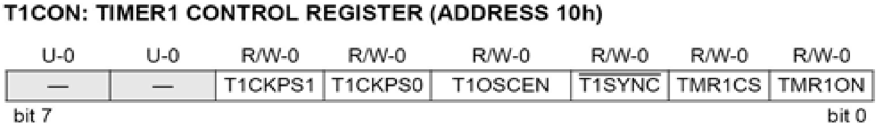
14. Give the timer 1 registers

- Timer1 has a register called TMR1 register, which is 16 bits of size.
- Actually, the TMR1 consists of two 8-bits registers:
- TMR1H
- TMR1L
- It increments from 0000h to the maximum value of 0xFFFFh (or 0 b1111 1111 1111 1111 or 65,535 decimal). The TMR1 interrupt, if enabled, is generated on overflow which is latched in interrupt flag bit, TMR1IF (PIR1<0>).

15. What is prescaler in timer1?

Prescaler can be used for further division of the system clock. The size of the register is 2-bit only, so four different division can be carried out The options are:
1:1
1:2
1:4
1:8

14. What is T1CON register ?

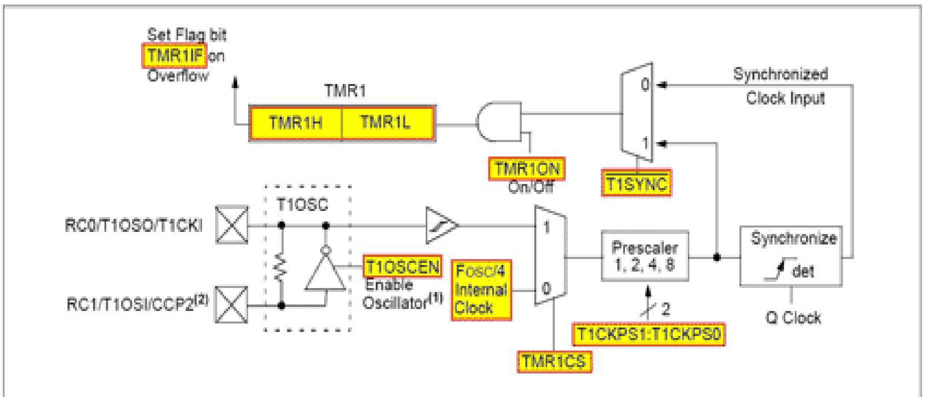


T1CON: TIMER1 CONTROL REGISTER (ADDRESS 10h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON

bit 7 bit 0

15. Give the block diagram of timer 1.



Note 1: When the T1OSCEN bit is cleared, the inverter is turned off. This eliminates power drain.

$$fout = \frac{fclk}{4 * \text{Prescaler} * (65536 - \text{TMR1}) * \text{Count}} \quad where \quad Tout = \frac{1}{fout}$$

16. Define subroutine? (Nov/Dec-2017)

- A subroutine is a coherent sequence of instructions that carries out a welldefined function and Conceptually, a subroutine is similar to a function call in a high-level language
- The same sequence of instructions can be used many times without the need to rewrite them over and over n Subroutines make programs easier to write (in a top-down fashion) and maintain.

17. Give the diagram of a state machine

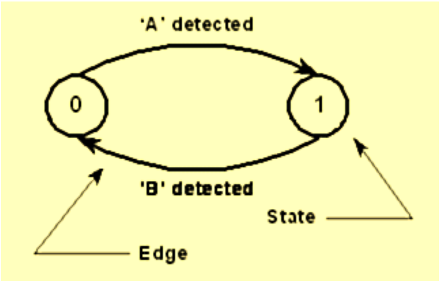


Figure 1. A simple state machine.

18. Brief the state machine model. (Nov/Dec-2017)

When an 'A' is detected in the stream in state 0, the machine makes a transition from state 0 to state 1, following the edge in the direction of the arrow. If a 'B' is detected in state 1, a transition is made to state 0. Since the state machine can only occupy one state at a time, the active state indicates whether the last character detected was either 'A' or 'B'. Conceivably, another character could be received ('C'), in which case no transition occurs.

19. What is key switch?

Push button switch is connected to the first bit of PORT D (RD0) which is configured as an input pin. Which is connected to a pull up resistor such that this pin is at Vcc potential when the switch is not pressed. When the switch is pressed this pin RD0 will be grounded. The LED is connected to the first bit of PORT B (RB0) and a resistor is connected in series with it to limit the current.

20. Write an ALP to initialize the PORT A using PIC microcontroller.(April/May-2017)

- BCF STATUS,RP1
- BCF STATUS,RP0
- CLRF PORT A

21. How to make LCD work with PIC?

```
void
lcd_puts(const char * s)
{
LCD_RS = 1;          // write characters
while(*s)
lcd_write(*s++);
}
```

22. What are the steps involved in writing a letter to the LCD display?

perform an initialization.
Send the desired position to IR (DDRAM Address**).**
Send ASCII code of the letter to DR.

23. What is GPIO with PIC microcontroller?

GPIO(General Purpose IO) is the most basic method of communication between MCU and external world. These are done with a PORT. Ports are nothing but a set of bits physically connected to PINs of Microcontroller and available outside the chip. the maximum size of a PORT is 8 bits. Some PORTs have fewer than 8 bits.

24. Give a program on ISR in PIC.

```
//Main Interrupt Service Routine (ISR)
      void interrupt ISR()
     {
 //Check if it is TMR0 Overflow ISR
 if (TMR0IE && TMR0IF)
     {
      //TMR0 Overflow ISR
      counter++;//Increment Over Flow Counter
     if(counter==76)
      {
```

```
      //Toggle RB1 (LED)
      if(RB1==0)
        RB1=1;
      else
        RB1=0;
      counter=0;  //Reset Counter
    }
    //Clear Flag
    TMR0IF=0;
  }
}
```

25. Assume that WREG has packed BCD. Write a program to convert packed BCD to two ASCII numbers.

```
MOVLW 0x29          ;       W = 29, packed BCD
ANDLW 0x0F          ;       Mask the upper nibble
IORLW 0x30          ;       Mask it an ASCII w=39
MOVWF 0x06
MOVLW 0x29
ANDLW 0xF0          ;       Mask the lower nibble (w=20H)
SWAPF WREG,W   ;    Swap the nibbles (w =02)
IORLW 0x30          ;       w=32
MOVWF 0x07
```

26. Multiply the following two unsigned bytes 81H and 04H and save the result in registers 10H and 11H respectively.

```
MOVLW 0X81
MULLW 0X04
MOVFF PRODL, 0X10
MOVFF PRODH, 9X11.
```

27. What is role of TRISX register? (Nov/Dec 2016)

It is called as data direction register which provides an access to the data flow through the respective ports initialized.

28. What is the minimum and maximum clock frequency of PIC16CXX?(Nov/Dec 2016)

It can operate from 1Mz to 33MHz.

29. Write an assembly language program to transfer data from memory block B1 to memory block B2?

```
MVI C,0AH; Initialize counter
LXI H, 2200H; Initialize source memory pointer
LXI D, 2300H; Initialize destination memory pointer
Loop:    MOV A,M; Get byte from source memory block
STAX D; Store byte in the destination memory block
INX H; Increment source memory pointer
INX D; Increment destination memory pointer
DCR C; Decrement counter
JNZ Loop ; If counter ¹ 0 repeat
HLT
```

30. Write an assembly language program to add 2 BCD numbers?

```
 LXI H,2200H; Initialize pointer
 MOV A,M ; Get the first number
 INX H; Increment the pointer
ADD M ; Add two numbers
```

DAA ; Convert HEX to valid BCD
STA 2300; store the result
HLT

PART-B (C406.2)

1. Define Interrupt & Explain the concepts of interrupts in detail. **(Nov/Dec 2016) & (Nov/Dec 2017), (April/May-2017)**
2. With a program, give the explanation of ISR in detail.
3. In a detail give an account of Timer programming RAM/ROM allocation in PC. **(Nov/Dec 2017)**
3. Explain timer 0 in detail with its registers.
4. Give the block diagram of timer 1 and its associated registers.
5. How timer2 is different from timer 0 and 1. Explain.
6. With a simple program explain the concept of timer in detail. **(Nov/Dec 2016)**
7. What is the value of count for a 0.5 second delay using timer 0?
8. Give a detailed note on state machine and key switches with a brief programming concept in PIC microcontroller.
9. Write a program to display a constant in PIC.
10. Write a program to create a delay of 1 sec using timer 0.
11. Explain the process and Procedure to display constant strings and variable strings. **(April/May-2017)**

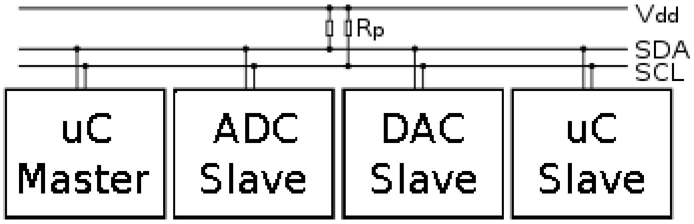
UNIT – III PERIPHERALS AND INTERFACING

PART – A

1. What is an I^{2}C bus?

I^{2}C is a serial protocol for two-wire interface to connect low-speed devices like microcontrollers, EEPROMs, A/D and D/A converters, I/O interfaces and other similar peripherals in embedded systems. It was invented by Philips and now it is used by almost all major IC manufacturers. I2C bus is popular because it is simple to use, there can be more than one master, only upper bus speed is defined and only two wires with pull-up resistors are needed to connect almost unlimited number of I2C devices. I2C can use even slower microcontrollers with general-purpose I/O pins since they only need to generate correct Start and Stop conditions in addition to functions for reading and writing a byte.

2. Draw the Block diagram of I^{2}C bus.



3. What are the modes of operation in I^{2}C bus?

There are three data transfer speeds for the I2C bus: standard, fast-mode, and high-speed mode. Standard is 100 Kbps. Fast-mode is 400 Kbps, and high-speed mode supports speeds up to 3.4 Mbps. All are backward compatible. The I2C bus supports 7-bit and 10-bit address space devices and devices that operate under different voltages.

4. How does PIC write data through I^{2}C bus?

- A peripheral chip address and a read/write bit designating that the peripheral chip is to read successive bytes.
- A peripheral internal register or address byte.
- Data to write into one or more consecutive internal addresses**.**

5. How PIC does reads data through I^{2}C bus?

- PIC sends out a peripheral chip address and a read/write bit designating that the peripheral chip is to send one or more successive bytes beginning at a previously selected internal register or address.

- Reads back one or more bytes of data.

6. Microntroller based control is more advantageous than conventional control-Justify.

- Faster operation
- Easily programmable and flexible
- Repetitive tasks can be easily accomplished
- Low cost

7. Write a subroutine program to intialize I^{2}C bus.

```
Movlw 00111011        ; Enable I2C Master mode.
Movwf SSPCON
bcf PORTC, SDA   ; DRIVE SDA low when it is an output
bcf PORTC, SCL   ; DRIVE SCL low when it is an output
Movlw TRISC  ; Set indirect pointer to TRISC
Movwf FSR
```

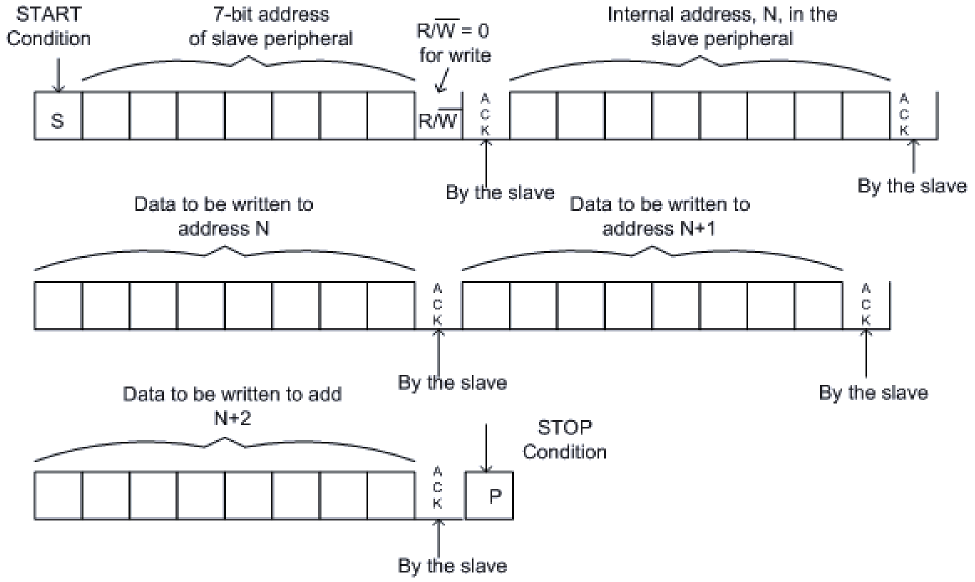
8. Difference between bus operation and bus subroutine. (Nov/Dec-2017)

Bus operation	**Bus subroutine**
It requires two open drain I/O pins called SCL(serial clock) and SDA(serial data).	SCL pin must have an open drain output while SDA pin must be either an input or have an open drain output.
Its can serve for connecting a master PIC to one or more slave PICs using two wires for connections. Eg: EEPROM,ADC,RAM ,LCD	It will repeatedly access TRISC, The data direction register for PORTC.
To change the output from 0V to high impedance output instead of writing a one to Port C bit , a One is written to corresponding TRISC bit.	TRISC is located at the bank 1 address; H'87' which cannot be accessed by direct addressing without first executing the instruction – Bsf STATUS ,PRO then changing the required bit of TRISC and finally reverting back to bank 0 with - Bcf STATUS , PRO

9. Calculate the resolution of 10bit ADC having Max. analog value +10.0 volts.

Resolution of a converter determines the degree of accuracy in conversion. It is equal to $1/2^n$so, $1/2^{10}$=0.000976.

10. Draw the format of I^{2}C bus to read and write from several peripheral interfaces.


START Condition
7-bit address of slave peripheral
R/W = 0 for write
Internal address, N, in the slave peripheral
S
R/W
ACK
ACK
By the slave
By the slave
Data to be written to address N
Data to be written to address N+1
ACK
ACK
By the slave
By the slave
Data to be written to add N+2
STOP Condition
ACK
P
By the slave

11. **How EEPROM memory stores the information?**

An EEPROM is a memory that allows storing the variables, as a result of burning the written program.

12. **Write a note on temperature sensor used for interfacing with I^{2}C bus.**

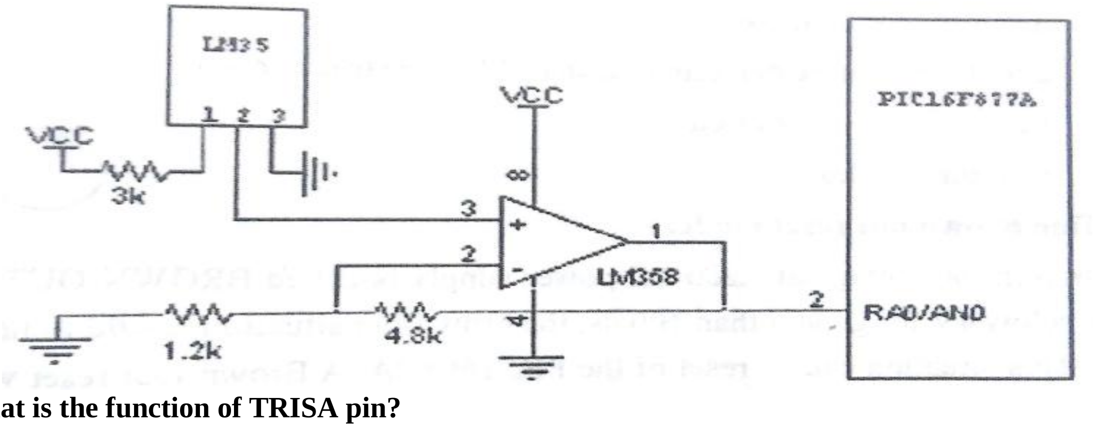
National Semiconductor's LM 75 chip combines an analog temperature transducer, an analog-to-digital convertor (9-bit), and an I^{2}C bus interface, all in a tiny S)-8 surface mount package. The temperature range covered is -25^{o}C to +100^{o}C with ±2^{o}C accuracy. The two's complement form of the temperature is available from the 9-bit ADC. The resolution of the ADC is about 0.5^{o}C.

13. **What are the features of 16C7X?**

Features (16C7X):

- Eight input channels
- An analog multiplexer
- A track and hold circuit for signal on the selected input channel
- Alternative clock sources for carrying out the conversion.
- An adjustable autonomous sampling rate.
- The choice of an internal or external ref. voltage.
- 8-bit conversion
- Interrupt response when conversion is completed.

14. **How is temperature sensor is interfaced with PIC microcontroller?(April/May 2017)**



15. **What is the function of TRISA pin?**

Setting TRISA bit will configure portA as input and resetting will configure as output port.

16. **What is the status of ADON?**

When ADON=0 then AD is off, when ADON=1 then AD is turned ON.

17. What are the bit positions of ADCON?

D0-ADON,D1-ADIF,D2-Go/Done,D3-CHSO,D4-CHS1,D5-undefined,D6-ADSC0,D7-ADSC1.

18. What is synchronous and asynchronous transmission?

Asynchronous – start and stop bit allowed for transmission of data. Synchronous – no start and stop bit only block header data.

19. What is baud rate in asynchronous mode?

The baud rate in asynchronous mode is given by B.R = Fosc/64.(x+1) for low speed, and Fosc/16(x+1) for high speed.

20. How data is transmitted serially using UART?

To transmit a byte of data serially from the TX pin, the byte is written to the TXREG register. Assuming there is not already data in the TSR, the content of TXREG will be automatically transferred to TSR, making TXREG available for a second byte even as the first byte is being shifted out of the TX pin, framed by START and STOP bits.

21. What is key debouncing?

Key bouncing may cause multiple entries made for the same key. To overcome this problem after a key press is sensed the device is made to wait for few milliseconds. Then the key is checked again to ensure it is still pressed. If it is still pressed it is taken as a valid key press. This process is called keyboard debouncing.

19. Name any two types of ADCs.

The different types of ADC are successive approximation ADC, Counter type ADC flash type ADC, integrator converters and voltage to-frequency converters.

20. What is the difference between A/D and D/A converters?

Digital-to-analog conversion is to pull the samples from memory and convert them into an impulse train. An ADC is attempting to capture and convert a largely unknown signal into a known representation. In contrast, a DAC is taking a fully known, well-understood representation and "simply" generating an equivalent analog value. The challenge for an ADC is much greater than it is for a DAC.

21. Define the following terms for D/A converters.

i) **Resolution: Resolution** of a converter determines the degree of accuracy in conversion. It is equal to 1/2n.

ii) Accuracy: Accuracy is the degree to which information on a map or in a digital database matches true or accepted values. Accuracy is an issue pertaining to the quality of data and the number of errors contained in a dataset or map. In discussing a GIS database, it is possible to consider horizontal and vertical accuracy with respect to geographic position, as well as attribute, conceptual, and logical accuracy.

iii) **Monotonicity :** If a clock has monotonicity, then each successive time reading from that clock will yield a time further in the future than the previous reading.

iv) **Conversion time:** The time required by an analog to digital converter to fully convert and analog input sample.

22. What are the functions of USART?

It converts parallel data to serial or vice versa. The data transmission or reception can be either synchronous or asynchronous. It can be used to interface MODEM and establish serial communication through MODEM over telephone line.

23. Write a program to initialize port A.

```
Org0
BcfSTATUS.RP0    clrf
PORTA
bsf STATUS.RP0
movlw 00010000H
```

movwf TRISA. End

24. What is the function of TRISA pin?

Setting TRISA bit will configure portA as input and resetting will configure as output port.

25. **How to initialize a delay in PIC?**

It is given by delay(100ms) which means delay of 100 ms is provided.

26. What is Left and Right justification in ADC?

There are 16 bits in the two result registers. The AD conversion takes 10 bits. Therefore, 6 bits are not used. The result format bit allows determining whether the first 6 bits of the high register are not used(right justified), or the six last bits of the low register are not used(left justified).

27. Write a program to transmit the message "YES" serially at 9600 baud, 8 bit data and 1 stop bit. Do this forever.

```
        MOVLW       B'0010000'
        MOVWF       TXSTA
        MOVLW       D'15'
        MOVWF       SPBRG
        BCF         TRISC, Tx
        BSF         RCSTA, SPEN
Over    MOVLW       A'Y'
        CALL        TRANS
        MOVLW       A'E'
        CALL   TRANS
        MOVLW       A'S'
        CALL        TRANS
        MOVLW       0x0
        CALL        TRANS
        BRA         Over

TRANS
S1      BTFSS       PIR1, TxIF
        BRA         S1
        MOVWF       TxREG
        RETURN
```

28. Define Baud Rate. (Nov/Dec-2017)

The **baud rate** is the **rate** at which information is transferred in a communication channel. In the serial port context, "9600 **baud**" means that the serial port is capable of transferring a maximum of 9600 bits per second.

28. What is the value to be loaded into SPBRG register if we want 19200 baud rate with 10MHZ clock source. (Nov/Dec 2016)

01101010 is the value to be loaded into SPBRG register

29. List the registers associated with UART. (Nov/Dec 2016) & (Nov/Dec-2017)

Register	**Description**
TXSTA	Transmit Status And Control Register
RCSTA	Receive Status And Control Register
SPBRG	USART Baud Rate Generator
TXREG USART	Transmit Register. Holds the data to to be transmitted on UART
RCREG	USART Transmit Register. Holds the data received from UART

30. What do you mean by Looping, Counting and Indexing?

- Looping: In this tech the program is instructed to execute certain set of instructions repeatedly toexecute a particular task number of times.
- Counting: This tech allows programmer to count how many times the ins of instruction are executed.

- Indexing: This tech allows programmer to point or refer the data stored in sequential memory location one by one.

PART –B

1. What is meant by I^2C module? Explain how I^2C is interfaced with PIC microcontroller. (**Nov/Dec 2016)**
2. Exhibit the operation of I2C bus and develop Embedded C program to transmit a data using I2C.**(Nov/Dec -2017)**
3. Explain with example the concept of I^2C subroutine in PIC microcontroller.
4. Illustrate with suitable example how I2C communication is carried out in PIC Microcontroller. **(April/May-2017)**
5. Explain in detail the interfacing of temperature sensor using I^2C bus.**(Nov/Dec -2017)**
6. Explain with neat diagram interfacing of serial EEPROM using I^2C bus.
7. Using suitable circuits, construct and explain how ADC is interfaced with PIC microcontroller. **(Nov/Dec 2016) & (Nov/Dec -2017), (April/May-2017)**
8. i) Explain in detail about Baud rate selection in UART.
   ii) Explain different types of errors that occur in Asynchronous serial transmission.
9. Explain with neat diagram the use of UART to interface two PIC resources.
10. With a neat sketch explain the concept of data handling and initialization of UART circuitry.
11. Explain with neat sketch how one or more channels can be periodically sampled using ADC.
12. Write a program in PIC for analog to digital conversion with ADCON and ADRES register.

UNIT – IV INTRODUCTION TO ARM PROCESSOR

PART-A

1. **Mention the key features of RISC architecture.**
    - A fixed instruction size with few formats
    - A load store architecture where instructions that process data operate only on registers and are separate from instructions that access memory.
    - A larger register bank of thirty two 32 bit registers, all of which could be used for any purpose, to allow the load store architecture to operate efficiently.
2. **Mention the advantages and disadvantages of RISC.**

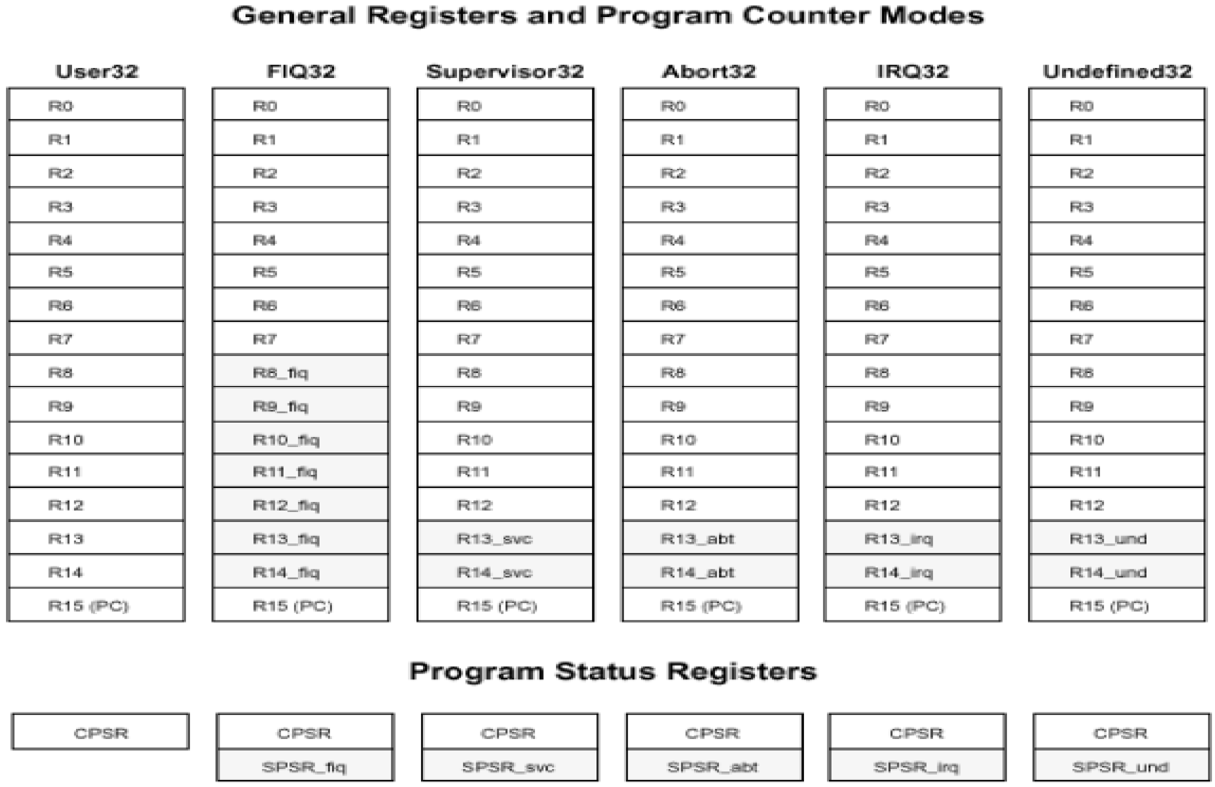
   Advantages:
    - Smaller die Size.
    - A shorter development time.
    - A higher performance.

   Disadvantages:
    - Poor code density compared to CISCs.
    - RISCs don't execute x86 code.
3. **What are the features of RISC architecture that were used in ARM architecture?**
    - A load store architecture
    - Fixed length 32 bit instructions
    - 3- Address instruction formats.
4. **What is CPSR?**

   The CPSR (Current Program Status Register) is used in user level programs to store the condition code bits. These bits are used for example, to record the result of a comparison operation and to control whether or not a condition branch is taken.
5. **What are the registers available in ARM processor?**
    - R0 to R15 directly accessible
    - R0-R14 general purpose

- R13 stack pointer
- R14 Linked register
- R15 holds Program Counter
- CPSR- Current Program Status Register contains condition code flags and current mode bits.
- 5 SPSRs (Saved Program Status Registers) which are loaded with CPSR when exception occurs.

6. Draw the ARM programmer's model.



General Registers and Program Counter Modes

User32	FIQ32	Supervisor32	Abort32	IRQ32	Undefined32
R0	R0	R0	R0	R0	R0
R1	R1	R1	R1	R1	R1
R2	R2	R2	R2	R2	R2
R3	R3	R3	R3	R3	R3
R4	R4	R4	R4	R4	R4
R5	R5	R5	R5	R5	R5
R6	R6	R6	R6	R6	R6
R7	R7	R7	R7	R7	R7
R8	R8_fiq	R8	R8	R8	R8
R9	R9_fiq	R9	R9	R9	R9
R10	R10_fiq	R10	R10	R10	R10
R11	R11_fiq	R11	R11	R11	R11
R12	R12_fiq	R12	R12	R12	R12
R13	R13_fiq	R13_svc	R13_abt	R13_irq	R13_und
R14	R14_fiq	R14_svc	R14_abt	R14_irq	R14_und
R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)

Program Status Registers

CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
	SPSR_fiq	SPSR_svc	SPSR_abt	SPSR_irq	SPSR_und

7. What are condition code flags in CPSR?

N: Negative; the last ALU operation which changed the flags produced a negative result(the top bit of the 32 bit result was a one).

Z: Zero; the last ALU operation which changed the flags produced a zero result (every bit of the 32 bit result was zero).

C: Carry; the last ALU operation which changed the flags generated a carry out, either as a result of an arithmetic operation in the ALU or from the shifter.

V: Overflow; the last arithmetic operation which changed the flags generated an overflow into the sign bit.

8. List out some features of ARM architecture.

- A large set of registers, all of which can be used for most purposes.
- 3- address instructions(that is, the two source operand registers and the result register are all independently specified)
- Conditional execution of every instruction.
- Inclusions of very powerful load and store multiple register instructions.

9. List out some ARM development tools. (Nov/Dec-2017)

ARM C compiler, ARM assembler, Linker, ARMsd, ARMulator, ARM development board.

10. What is ARMulator? (April/May-2017)

The ARMulator is a suite of programs that models the behavior of various ARM processor cores in software on a host system.

11. What are the various levels of accuracy in ARMulator? (April/May-2017)

Instruction accurate modeling gives the exact behavior of the system state without regard to the precise timing characteristics of the processor.

Cycle accurate modeling gives the exact behavior of the processor on a cycle by cycle basis, allowing the exact number of clock cycles that a program requires to be established.

Timing accurate modeling presents signals at the correct time within a cycle, allowing logic delays to be accounted for.

12. How exceptions are handled in ARM processor?

The current state is saved by copying the PC into r14_exc and the CPSR into SPSR_exc

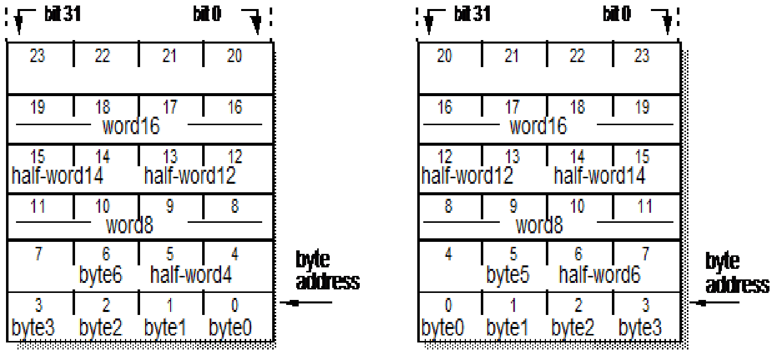
The processor operating mode is changed to the appropriate exception mode.

The PC is forced to a value between 00_{16} and $1C_{16}$, the particular value depending on the type of exception.

13. Define Context Switching.(April/May-2017)

Switching of the CPU of a microcontroller from one process or thread to another process or thread.

14. Draw Memory organization in ARM processor.



(a) Little-endian memory organization

(b) Big-endian memory organization

15. What are the seven modes of operations in ARM processor?

User mode(usr),Fast interrupt mode(fiq), Interrupt mode(irq), supervisor mode(svc),abort(abt),,system(sys), undefined mode(und).

16. List out the types of instructions used in ARM processor.

i. Data processing instructions

ii. Data Transfer instructions.

iii. Control flow instructions.

17. What is meant by load store architecture?

Load store architecture means that the instruction set will only process values which are in registers and will always place the results of such processing into a register. The only operations which apply to memory state are ones which copy memory values into registers or copy register values into memory.

18. What are the rules that apply to Data Processing instructions?

- All operands are 32 bits wide and come from registers or are specified as literals in the instruction itself.
- The result, if there is one, is 32 bit wide and is placed in a register.
- Each of the operand registers and the result register are independently specified in the instruction. That is, the ARM uses a 3- address format for these instructions.

19. Mention the data processing instructions used in ARM processor.

Arithmetic operations, Bit wise logical operations, Register movement operations, Comparison operations.

20. What are the basic forms of Data transfer instructions?

- Single register load and store instructions.
- Multiple register load and store instructions
- Single register swap instructions.

21. What is register indirect addressing? Give example.

Register indirect addressing uses a value in one register (the base register)as a memory address and either loads the value from that address into another register or stores the value from another register into that memory address.

e.g. LDR r0, [r1] ; r0 : = mem_{32}[r1]
STR r0, [r1] ; mem_{32}[r1] : =r0

22. What is pre indexed and post indexed addressing modes?

The preindexed addressing mode is one which allows one base register to be used to access a number of memory locations which are in the same area of memory.

The post indexed addressing, allows the base to be used without an offset as the transfer address, after which it is auto indexed.

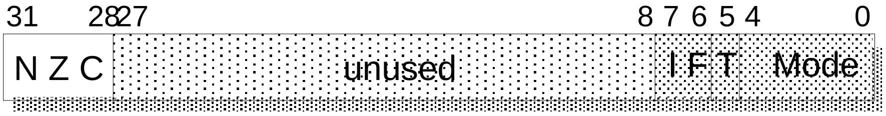
23. What are the forms of stack supported by ARM multiple transfer instructions?

Full ascending, empty ascending, full descending, empty descending.

24. Mention the features of Memory organization of ARM processor.

Byte addressable, 32 bit address space
Little or Big endian addressable
32 bit word length
Word, half word and byte data transfers to and from processor registers.
Word and half word transfers must be aligned.

25. Draw CPSR format of ARM processor.



26. What is cache memory?

A cache memory is a small, very fast memory that retains copies of recently used memory values. It operates transparently to the programmer, automatically deciding which values to keep and which to overwrite.

27. What are the advantages and disadvantages of onchip RAM over cache memory?

Advantages:
It is simpler, cheaper and uses less power.
It has more deterministic behavior.
Enables programmer to allocate space in it using knowledge of the future processing load.
Disadvantages:
It requires explicit management by the programmer, whereas cache is usually transparent to the programmer.

28. Define unified and Harvard cache.

A unified cache is a single cache for both instructions and data
Modified Harvard organization uses separate instruction and data caches.

29. Define hit and miss rate.

The proportion of all the memory accesses that are satisfied by the cache is the hit rate, usually expressed as a percentage and the proportions that are not is the miss rate.

30. What are the two memory management approaches?

The two memory management approaches are segmentation and paging.

31. What is paging?

In paging memory management scheme both the logical and the physical address spaces are divided into fixed size components called pages. A page is usually a few kilobytes in size, but different architectures use different page sizes. The relationship between the logical and physical pages is stored in page tables, which are held in main memory.

32. Give the CP15 protection unit register structure.

Register	Purpose
0	ID register
1	Configuration
2	Cache control
3	Write Buffer control
5	Access Permissions
6	Register Base and size
7	Cache Operations
9	Cache Lock Down
15	Test
4,8,10-14	unused

33. What are the commonly used write strategies in fully associative cache?

The commonly used write strategies are

- Write through
- Write through with buffered write
- Copyback

34. What is the purpose of Program Counter? (**Nov/Dec 2016)**

1. Store the address of the instruction to be executed.
2. All instructions are 32 bit wide
3. Thus, the last 2 bits of PC are undefined.

35. List out some ARM Development tools.(Nov/Dec 2016)

Keil-C, IAR bench, mbed, ARM DS-5 development studio are some of the ARM development tools.

PART-B

1. With Neat sketch explain the functional block diagram of ARM architecture. (**Nov//Dec 2016) & (Nov/Dec 2017)**
2. Explain the various operating modes programmers model in ARM processor.(**Nov//Dec 2016), (April/May-2017)**
3. Write the subroutine program to output a text string following a CALL instruction using ARM processor. **(April/May-2017)**
4. Explain in detail ARM development tools.
5. (i)Explain the various data operations involved in ARM.**(Nov/Dec 2017)**
(ii)Illustrate the concept of data operations in ARM processors.
6. Explain with examples in detail the Data processing instructions of ARM processor.
7. Explain with examples in detail the Data transfer instructions of ARM processor.
8. Explain with examples in detail the Control flow instructions of ARM processor
9. Explain with examples different types of addressing used in ARM processor.
10. i. Write and ARM ALP to display a text “Hello World” .
ii. Write and ARM ALP which dumps a register to the display in hexadecimal notation.
11. Explain two way set associative cache memory organization in ARM processor with neat sketch.
12. Explain with neat sketch the fully associative cache memory organization.
13. Explain in detail with neat sketch segmentation and paging in Memory organization.
14. Explain with neat sketch the operation of translation look aside buffer.
15. Explain in detail CP15 protection unit registers.
16. Write short notes on ARM MMU architecture.

UNIT – V ARM ORGANIZATION

PART-A

1. What is instruction pipelining?

Instruction pipelining is a technique that implements a form of parallelism called instruction-level parallelism within a single processor. It therefore allows faster CPU throughput (the number of instructions that can be executed in a unit of time) than would otherwise be possible at a given clock rate. The basic instruction cycle is broken up into a series called a pipeline. Rather than processing each instruction sequentially (finishing one instruction before starting the next), each instruction is split up into a sequence of steps so different steps can be executed in parallel and instructions can be processed concurrently (starting one instruction before finishing the previous one).

2. What does classic RISC pipeline comprises of?

The classic RISC pipeline comprises:

- Instruction fetch
- Instruction decode and register fetch Execute
- Memory access
- Register write back

3. What are the three processes used by pipelining technique?

Pipelined processors commonly use three techniques to work as expected when the programmer assumes that each instruction completes before the next one begins:

Processors that can compute the presence of a hazard may stall, delaying processing of the second instruction (and subsequent instructions) until the values it requires as input are ready. This creates a bubble in the pipeline (see below), also partly negating the advantages of pipelining.

Some processors can not only compute the presence of a hazard but can compensate by having additional data paths that provide needed inputs to a computation step before a subsequent instruction would otherwise compute them, an attribute called operand forwarding.

Some processors can determine that instructions other than the next sequential one are not dependent on the current ones and can be executed without hazards. Such processors may perform out-of-order execution.

4. What are the special situations in a pipelining structure?

Self-modifying programs

The technique of self-modifying code can be problematic on a pipelined processor. In this technique, one of the effects of a program is to modify its own upcoming instructions. If the processor has an instruction cache, the original instruction may already have been copied into a prefetch input queue and the modification will not take effect.

Uninterruptible instructions

An instruction may be uninterruptible to ensure its atomicity, such as when it swaps two items. A sequential processor permits interrupts between instructions, but a pipelining processor overlaps instructions, so executing an uninterruptible instruction renders portions of ordinary instructions uninterruptible too. The Cyrix coma bug would hang a single-core system using an infinite loop in which an uninterruptible instruction was always in the pipeline.

5. What are the design considerations in pipelining?

Speed

Pipelining keeps all portions of the processor occupied and increases the amount of useful work the processor can do in a given time. Pipelining typically reduces the processor's cycle time and increases the throughput of instructions. The speed advantage is diminished to the extent that execution encounters hazards that require execution to slow below its ideal rate. A non-pipelined processor executes only a single instruction at a time. The start of the next instruction is delayed not based on hazards but unconditionally.A pipelined processor's need to organize all its work into modular steps may require the duplication of registers that increases the latency of some instructions.

Economy

By making each dependent step simpler, pipelining can enable complex operations more economically than adding complex circuitry, such as for numerical calculations. However, a processor that declines to pursue increased speed with pipelining may be simpler and cheaper to manufacture.

Predictability

Compared to environments where the programmer needs to avoid or work around hazards, use of a non-pipelined processor may make it easier to program and to train programmers. The non-pipelined processor also makes it easier to predict the exact timing of a given sequence of instructions.

6. Give the generic pipeline with four stages.

- Fetch
- Decode
- Execute
- Write-back

7. Give the 3-stage ARM pipeline. (Nov/Dec 2017)

Fetch:

• The instruction is fetched from memory decode

• The instruction is decoded and the datapath control signals prepared for the next cycle\

Decode:

The instruction is decoded and register operands read from the register file. There are three operand read ports in the register file, so most ARM instructions can source all their operands in one cycle.

Execute:

• The operands are read from the register bank, shifted, combined in the ALU and the result written back

8. Give the 3-stage ARM pipeline organization. (Nov/Dec 2017)

ARM components:

• register bank

– 2 read ports, 1 write port

– Plus additional read and write ports for r15

• Barrel shifter

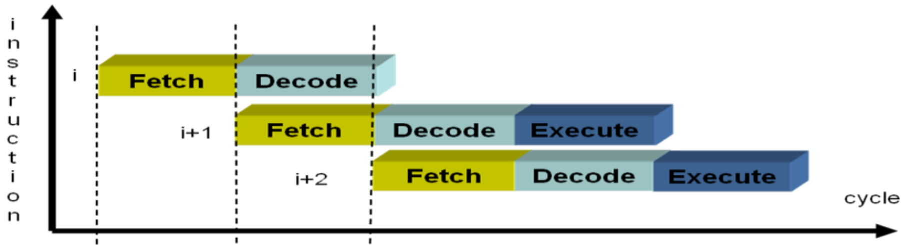
• ALU

• address register and incrementer

• Memory data registers

• Instruction decoder and control

9. Give the diagrammatic view of 3-stage pipeline.



10. What is 5-stage ARM pipeline organization?

Reduces work per cycle => allows higher clock frequency

Separates data and instruction memory => reduction of CPI

(Average number of clock Cycles Per Instruction

11. What are the operating modes?

Seven operating modes:

- User

- Privileged:
- System (version 4 and above)
- FIQ
- IRQ
- Abort
- Undefined
- Supervisor

12. What is user mode and exception mode?

User mode:

- Normal program execution mode
- System resources unavailable
- Mode changed by exception only

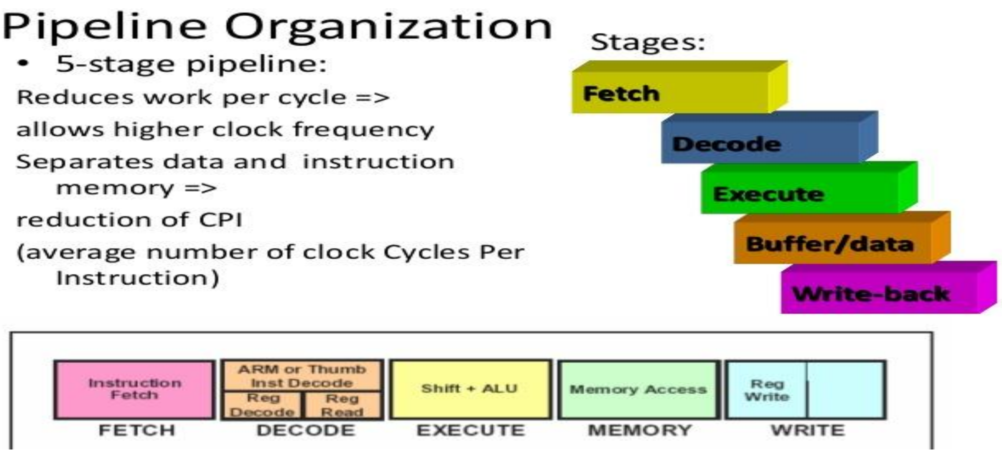
Exception modes:

- Entered upon exception
- Full access to system resources
- Mode changed freely.

13. What do you mean by control stalls?

Branches often introduce stalls (branch penalty)Stall time may depend on whether branch is taken ¾ May have to squash instructions that already started executing ¾ doesn't know what to fetch until condition is evaluated.

14. Give the diagram of 5-stage pipeline organization.



Pipeline Organization

- 5-stage pipeline:

Reduces work per cycle =>
allows higher clock frequency
Separates data and instruction memory =>
reduction of CPI
(average number of clock Cycles Per Instruction)

15. What is Tprog in 5-pipeline?

Tprog = Ninst * CPI / fclk

Where Ninst = no. of ARM instructions executed

CPI = average no. of clock cycles per instruction.

Fclk – clock frequency

16. What is five stage pipeline in ARM processor? (Nov/Dec 2016)

A typical 5-stage ARM pipeline is that employed in the ARM9TDMI.

The ARM processors which use a 5-stage pipeline have the following pipelinestages:

• Fetch:

the instruction is fetched from memory and placed in the instruction pipeline.

• Decode:

The instruction is decoded and register operands read from the register file. There are three operand read ports in the register file, so most ARM instructions can source all their operands in one cycle.

• Execute;an operand is shifted and the ALU result generated. If the instruction is a load or store the memory address is computed in the ALU.

• Buffer/data;data memory is accessed if required. Otherwise the ALU result is simply buf-fered for one clock cycle to give the same pipeline flow for all instructions

• Write-back;the results generated by the instruction are written back to the register file,including any data loaded from memory.

17. What are the main features of ARM instruction set?

- All instructions are 32 bits long.
- Most instructions execute in a single cycle.
- Every instruction can be conditionally executed.
- Load/store architecture
- Data processing instructions act only on registers
- Three operand format
- Combined ALU and shifter for high speed bit manipulation
- Specific memory access instructions with powerful auto-indexing addressing modes.
- 32 bit and 8 bit data types and also 16 bit data types on ARM Architecture v4.
- Flexible multiple register load and store instructions
- Instruction set extension via coprocessors

18. Give the registers of ARM.

ARM has 37 registers in total, all of which are 32-bits long.
1 dedicated program counter
1 dedicated current program status register
5 dedicated saved program status registers
30 general purpose registers

19. What is stack in ARM?

A stack is an area of memory which grows as new data is “pushed” onto the “top” of it, and shrinks as data is “popped” off the top.
Two pointers define the current limits of the stack.
A base pointer ,used to point to the “bottom” of the stack (the first location).
A stack pointer, used to point the current “top” of the stack.

20. What is co-processor interface in ARM processor?

The processor supports the connection of on-chip coprocessors through an external coprocessor interface. All types of coprocessor instruction are supported.
The ARM instruction set supports the connection of 16 coprocessors, numbered 0-15, to an ARM processor. In the processor, the following coprocessor numbers are reserved:
CP10
VFP control
CP11
VFP control
CP14
Debug and ETM control
CP15

21. How does the coprocessor interface of the ARM7TDMI work?

The coprocessor has to follow the pipeline of the ARM7TDMI. So it must have 3 stages (fetch, decode & execute), each holding one ARM instruction. The pipeline will advance each time the ARM does an instruction fetch, so the coprocessor pipeline stage will be controlled by (ECLK and NOT(nOPC)). At the decode stage of the pipeline, the coprocessor should examine the instruction opcode it has fetched. If it is a coprocessor instruction that it recognizes, it must look to see if the nCPI ARM output goes low in the execution stage - if so, then the coprocessor instruction should be executed.

22. What is the role of a Co-Processor?

Coprocessor provides functionalities that the main processor doesn’t provide.For example 8087 is numeric Coprocessor interfaced to 8086 mircoprocessor.8086 microprocessor performs integer(fixed point) arithmetic.8087 performs floating point arithmetic.

22. List few embedded applications for ARM processor. (Nov/Dec 2016), (Nov/Dec 2017)

Smart card, in-vehicle infotainment, Smart meters, Robotics,etc are some of the applications.

23. What does cortex-M3 processor brings in ARM applications?

The Cortex-M3 processor brings together multiple technologies to reduce memory size while delivering industry-leading performance in a small power efficient RISC core and delivers an ideal platform to accelerate the migration of thousands of applications around the globe from legacy components to 32-bit microcontrollers.

24. What is debugging in ARM?

ARM processors include hardware debugging facilities, allowing software debuggers to perform operations such as halting, stepping, and breakpointing of code starting from reset. These facilities are built using JTAG support, though some newer cores optionally support ARM's own two-wire "SWD" protocol. In ARM7TDMI cores, the "D" represented JTAG debug support, and the "I" represented presence of an "Embedded ICE" debug module. For ARM7 and ARM9 core generations, Embedded ICE over JTAG was a de facto debug standard, though not architecturally guaranteed.

25. Give an assembly language module in ARM.

```
AREA    ARMex, CODE, READONLY
                        ; Name this block of code ARMex
      ENTRY           ; Mark first instruction to execute
start
      MOV     r0, #10       ; Set up parameters
      MOV     r1, #3
      ADD     r0, r0, r1    ; r0 = r0 + r1
stop
      MOV     r0, #0x18     ; angel_SWIreason_ReportException
      LDR     r1, =0x20026  ; ADP_Stopped_ApplicationExit
      SVC     #0x123456     ; ARM semihosting (formerly SWI)
      END                   ; Mark end of file
```

26. What is the need of Thumb instruction set in ARM processor?

The Thumb instruction set is a subset of the most commonly used 32-bit ARM instructions. Thumb instructions are each 16 bits long, and have a corresponding 32-bit ARM instruction that has the same effect on the processor model. Thumb instructions operate with the standard ARM register configuration, allowing excellent interoperability between ARM and Thumb states.

27. **What is memory mapping? (may/june 2011)**

Memory mapping may refer to: Memory-mapped file, also known as map()MemorymappedI/O, an alternative to port I/O; a communication between CPU and peripheral device using the sameinstructions, and same bus, as between CPU and memory Virtual memory, technique which gives anapplication program the impression that it has contiguous working memory, while in fact it is physicallyfragmented and may even overflow on to disk storage.

28. **What are the significance of SFRs?(NOV 2010)**

SFRs denotes Special function Registers of 8051 controller. All the controller registers such as port latches, timer register, peripheral control register, accumulator, PC and DPTR all are available in SFR region.

29. **What is a Data pointer register?**

The data pointer register (DPTR) consists of a high byte(DPH) and a low byte (DPL) functions to hold 16 bit address. It may be manipulated as a 16-bit data register or as independent 8-bit registers. It serves as a base register in indirect jumps, look up table instructions and external data transfer.

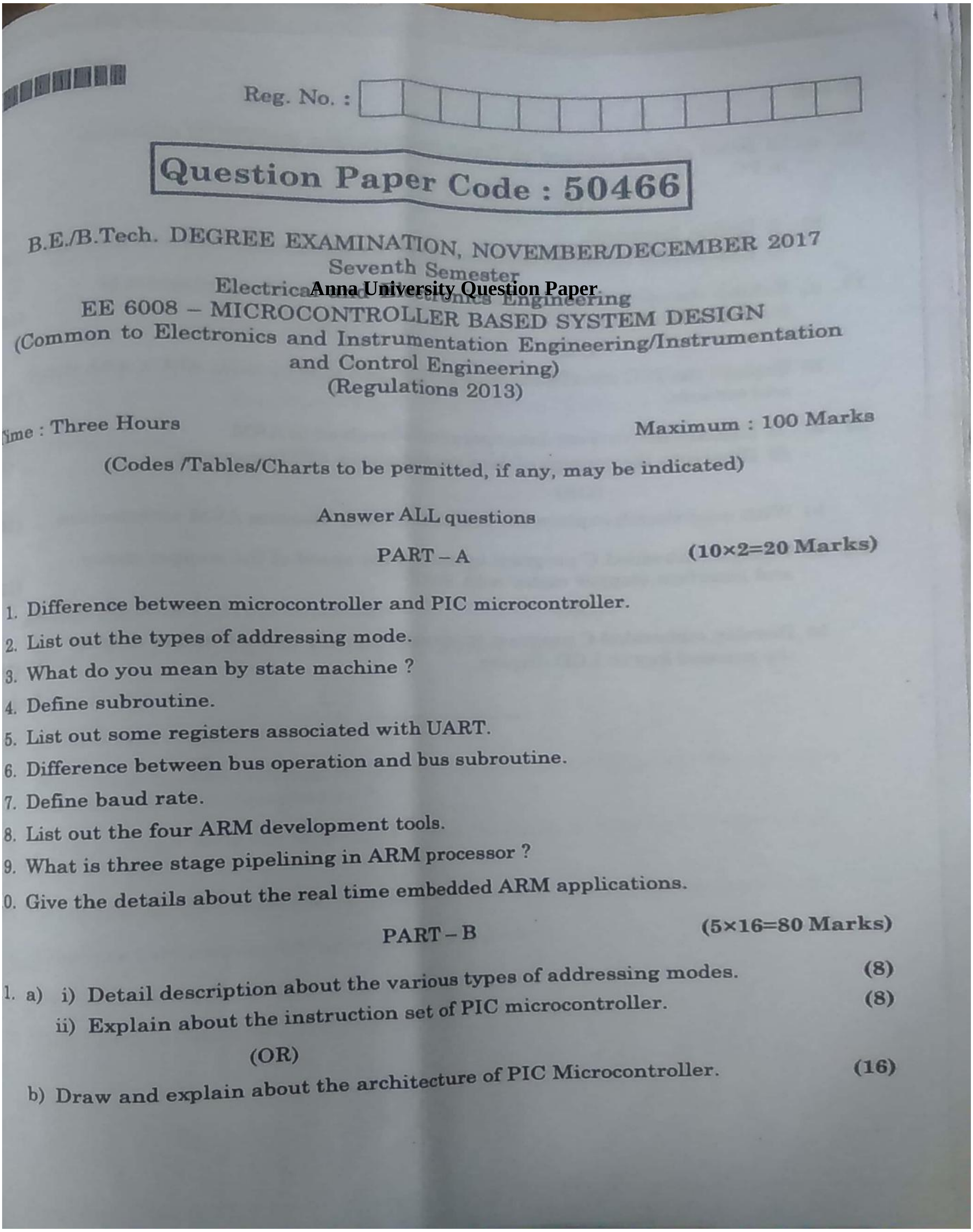
30. **What is SCON?**

SCON is the serial port control register , which contains not only the mode selection bits (SM0 – SM2 ,REN), but also the 9th data bit for transmit and receive (TB8 and RB8) and the serial port interrupt bits (TI and RI).

SM0 – Serial port mode control bit 0
SM1 – Serial port mode control bit 1
SM2 – Serial port mode control bit 2
REN – Receiver enable control bit
TB8 – Transmit bit 8
RB8 – Receive bit 8
TI – Transmit Interrupt flag
RI – Receive interrupt flag

PART-B

1. Give a detailed note on 3-state ARM pipeline organization. Show the difference between a single cycle and a multi-cycle instruction
2. Explain 5-stage ARM pipeline organization in detail. Explain the concept of data forwarding in this architecture. **(April/May-2017)**
3. Compare 3-stage and 5-stage ARM pipeline organization
4. Explain about SMART CARD system in detail.
5. What is In-vehicle infotainment using ARM? Explain.
6. Explain the history of ARM implementation.
7. Using suitable examples, explain the various instruction sets of ARM processor **(Nov/Dec 2016)**
8. Explain the co-processor interface of ARM in detail. **(Nov/Dec 2016)**
9. Write a program using data transfer and loading instruction in ARM processor.
10. Describe the ARM instruction execution mechanism showing the datapath activity for (a) Data processing instructions (b) Data transfer instructions (c) Branch instructions.
11. Write embedded C program to control the speed of the stepper motor and interface stepper motor with 8051.**(Nov/Dec 2017)**
12. Develope embedded C program to identify the key pressed and to display the pressed key in LCD display. **(Nov/Dec 2017)**
13. Elaborate the working principle of VLSI ISDN subscriber processor in detail. **(April/May-2017)**



Reg. No. : □□□□□□□□□□□□

Question Paper Code : 50466

B.E./B.Tech. DEGREE EXAMINATION, NOVEMBER/DECEMBER 2017

Seventh Semester

Electrical and Electronics Engineering

EE 6008 – MICROCONTROLLER BASED SYSTEM DESIGN

(Common to Electronics and Instrumentation Engineering/Instrumentation and Control Engineering)

(Regulations 2013)

ime : Three Hours Maximum : 100 Marks

(Codes /Tables/Charts to be permitted, if any, may be indicated)

Answer ALL questions

PART – A (10×2=20 Marks)

1. Difference between microcontroller and PIC microcontroller.
2. List out the types of addressing mode.
3. What do you mean by state machine ?
4. Define subroutine.
5. List out some registers associated with UART.
6. Difference between bus operation and bus subroutine.
7. Define baud rate.
8. List out the four ARM development tools.
9. What is three stage pipelining in ARM processor ?

0. Give the details about the real time embedded ARM applications.

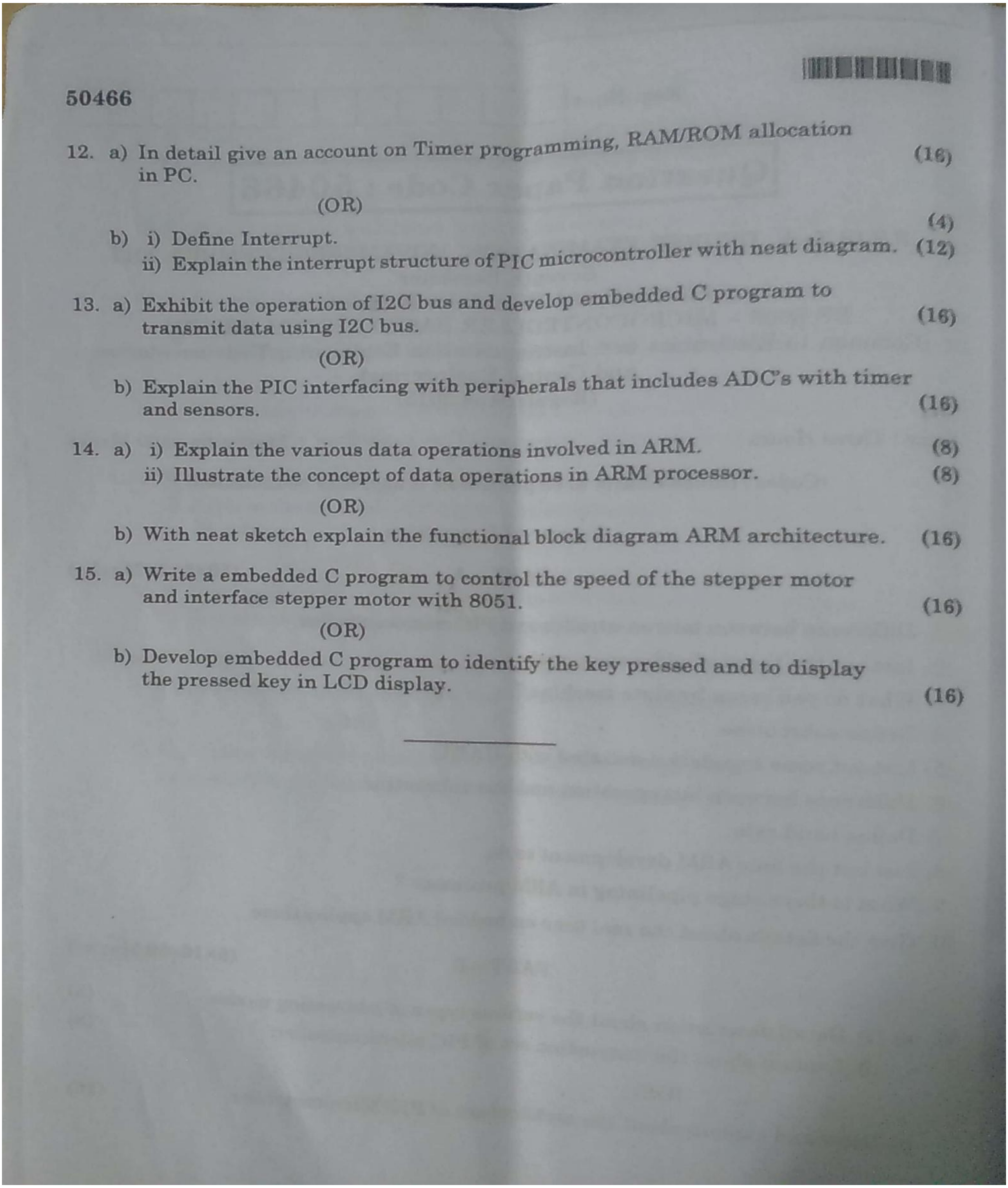
PART – B (5×16=80 Marks)

1. a) i) Detail description about the various types of addressing modes. (8)

ii) Explain about the instruction set of PIC microcontroller. (8)

(OR)

b) Draw and explain about the architecture of PIC Microcontroller. (16)



50466

12. a) In detail give an account on Timer programming, RAM/ROM allocation in PC. (16)

(OR)

b) i) Define Interrupt. (4)

ii) Explain the interrupt structure of PIC microcontroller with neat diagram. (12)

13. a) Exhibit the operation of I2C bus and develop embedded C program to transmit data using I2C bus. (16)

(OR)

b) Explain the PIC interfacing with peripherals that includes ADC's with timer and sensors. (16)

14. a) i) Explain the various data operations involved in ARM. (8)

ii) Illustrate the concept of data operations in ARM processor. (8)

(OR)

b) With neat sketch explain the functional block diagram ARM architecture. (16)

15. a) Write a embedded C program to control the speed of the stepper motor and interface stepper motor with 8051. (16)

(OR)

b) Develop embedded C program to identify the key pressed and to display the pressed key in LCD display. (16)

———————

Reg. No. : □□□□□□□□□□□□

Question Paper Code : 80363

B.E./B.Tech. DEGREE EXAMINATION, NOVEMBER/DECEMBER 2016.

Seventh Semester

Electrical and Electronics Engineering

EE 6008 — MICROCONTROLLER BASED SYSTEM DESIGN

(Common to Electronics and Instrumentation Engineering and Instrumentation and Control Engineering)

(Regulations 2013)

Time : Three hours

Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Write about the Status Register of PIC Microcontroller.

2. List out all the addressing Modes in PIC Microcontroller.

3. What is the minimum and maximum clock frequency for PIC 16CXX?

4. What is the role of TRISx register in I/O Port Management?

5. What is the value to be loaded into SPBRG register if we want 19200 baud rate with 10MHz clock source.

6. List the registers associated with UART.

7. What is the purpose of Program Counter?

8. List out some of ARM Development Tools.

9. What is five stage pipeline in ARM PROCESSOR?

10. List few embedded Application for ARM processor.

PART B — (5 × 16 = 80 marks)

11. (a) (i) Draw and explain the architecture of PIC 16 Microcontroller. (10)

(ii) Explain about the instruction set of PIC Microcontroller. (6)

Or

(b) Explain about the Various Memory organization of PIC Microcontroller. (16)

12. (a) Explain the functionality of TIMER for PIC Microcontroller with a suitable program. (16)

Or

(b) What is Interrupt? Explain the Interrupt structure of PIC Microcontroller with neat diagram. (16)

13. (a) What is meant by I^2C module? Explain how I^2C is interfaced with PIC Microcontroller. (16)

Or

(b) Using Suitable circuits, construct and explain how ADC is interfaced with PIC microcontroller. (16)

14. (a) With Neat sketch explain the functional block diagram ARM architecture. (16)

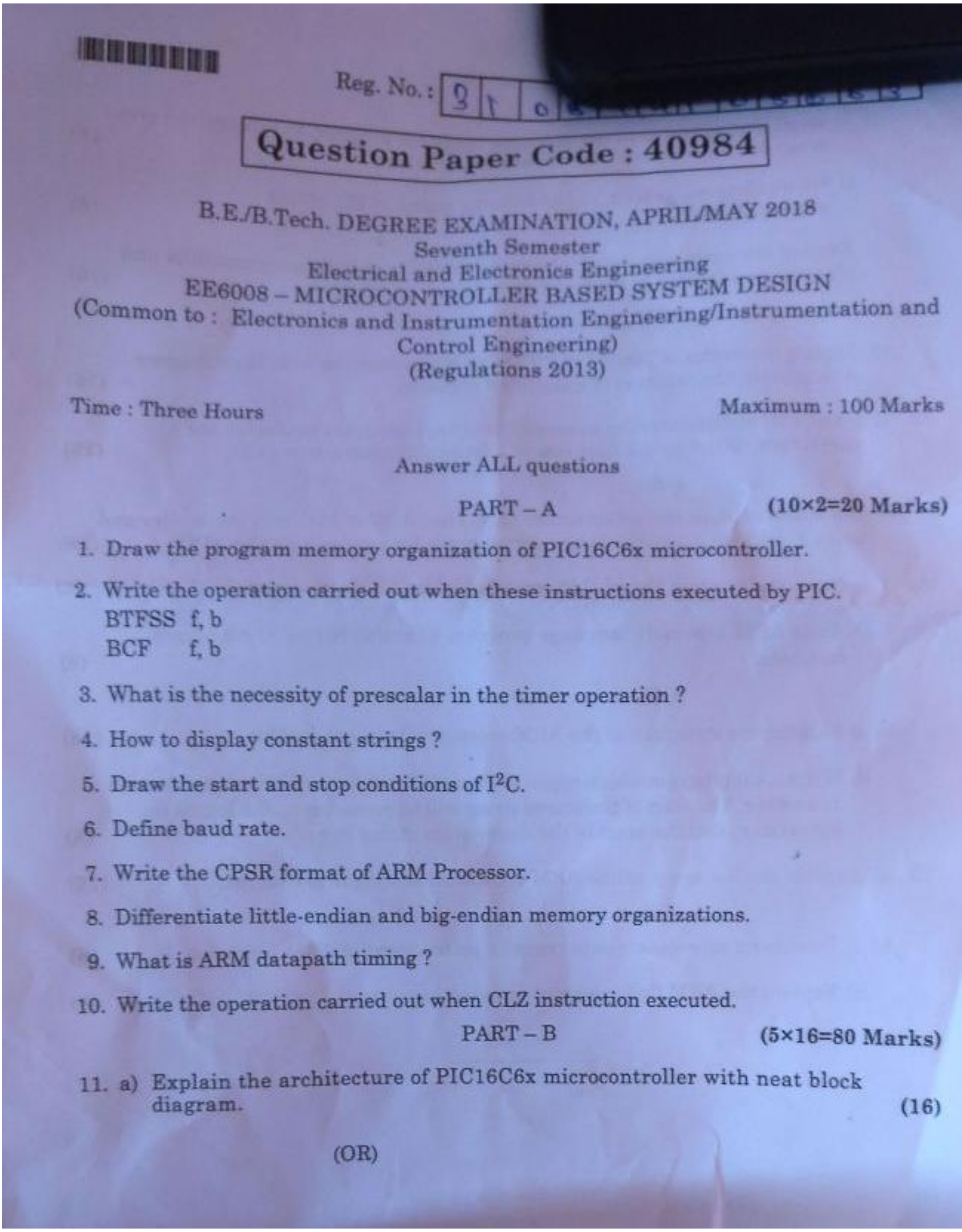
Or

(b) Explain the various Operating modes Programmers model in Arm Processor. (16)

15. (a) Using Suitable example, explain the various instruction set of ARM processor. (16)

Or

(b) Explain how does the coprocessor interface of the ARM work. (16)



Reg. No. : 9 1 0 [redacted]

Question Paper Code : 40984

B.E./B.Tech. DEGREE EXAMINATION, APRIL/MAY 2018

Seventh Semester

Electrical and Electronics Engineering

EE6008 – MICROCONTROLLER BASED SYSTEM DESIGN

(Common to : Electronics and Instrumentation Engineering/Instrumentation and Control Engineering)

(Regulations 2013)

Time : Three Hours Maximum : 100 Marks

Answer ALL questions

PART – A (10×2=20 Marks)

1. Draw the program memory organization of PIC16C6x microcontroller.

2. Write the operation carried out when these instructions executed by PIC.
   BTFSS f, b
   BCF f, b

3. What is the necessity of prescalar in the timer operation ?

4. How to display constant strings ?

5. Draw the start and stop conditions of I^2C.

6. Define baud rate.

7. Write the CPSR format of ARM Processor.

8. Differentiate little-endian and big-endian memory organizations.

9. What is ARM datapath timing ?

10. Write the operation carried out when CLZ instruction executed.

PART – B (5×16=80 Marks)

11. a) Explain the architecture of PIC16C6x microcontroller with neat block diagram. (16)

(OR)

b) i) Write PIC microcontroller assembly language program to arrange the given array having byte type data in ascending order. (8)

ii) With examples, explain the addressing modes of PIC16C6x microcontroller. (8)

12. a) Explain the various types of interrupts available in PIC microcontroller and also the step-by-step procedure to process an interrupt. (16)

(OR)

b) Explain the modes of Timer 1 of PIC16C6x microcontroller with block diagram. Also explain the function of associated registers. (16)

13. a) Write PIC microcontroller assembly language program to display the characters '2018' in the first row of 2 lines × 20 characters LCD. (16)

(OR)

b) Draw and explain the architecture of on chip ADC of PIC microcontroller and write a suitable assembly language program for configuring the ADC. (16)

14. a) i) Draw and explain the visible registers in an ARM processor. (8)

ii) Write ARM assembly language program to multiply two 32-bit binary numbers. (8)

(OR)

b) i) Explain the structure of the ARM cross-development tool kit. (8)

ii) Write a subprogram which copies a string of bytes from one memory location to another. The start of the source string will be passed in r_1, the length (in bytes) in r_2 and the start of the destination string in r_3. (8)

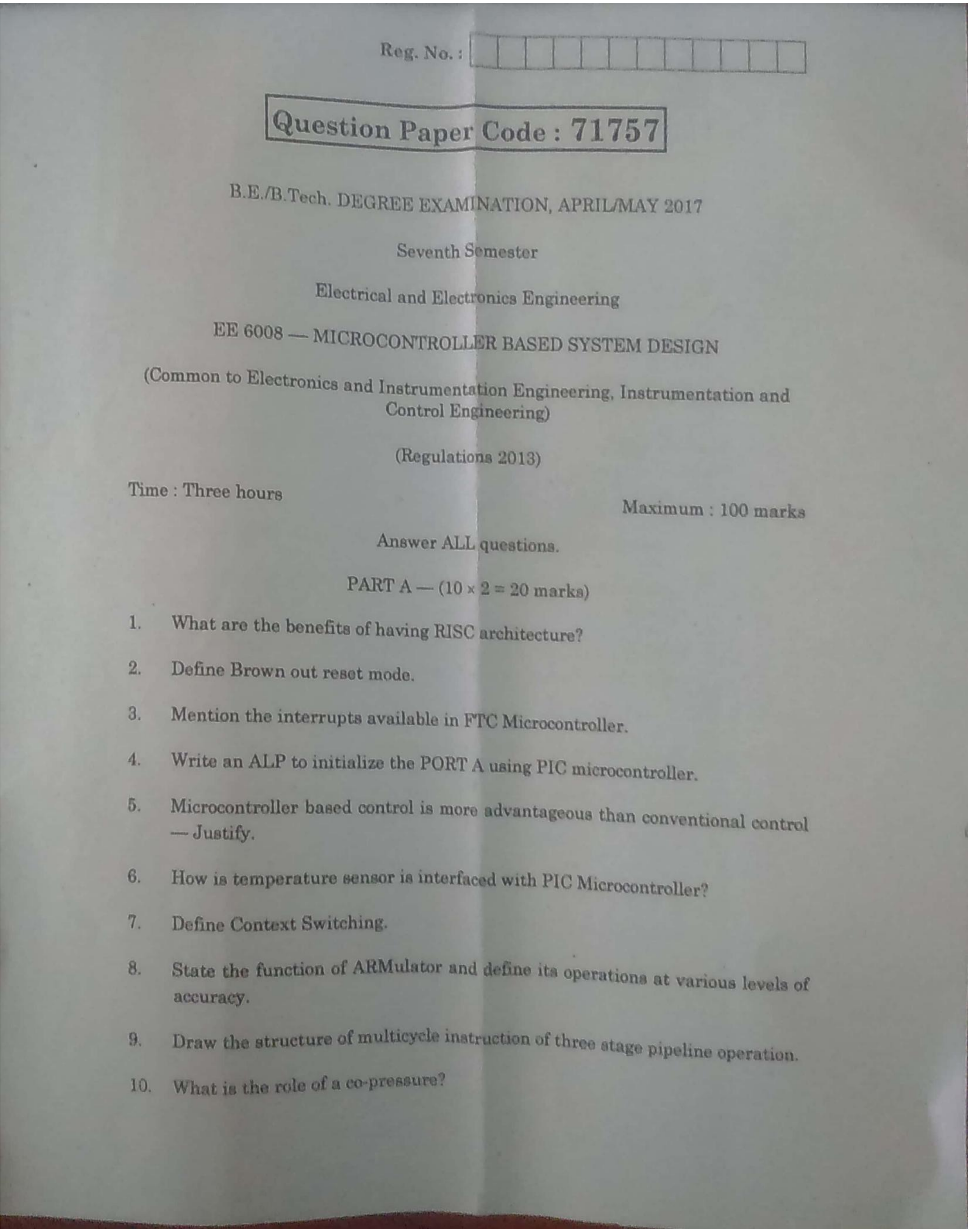
15. a) Explain the 5-stage pipeline ARM organization with neat diagram. (16)

(OR)

b) i) Discuss on coprocessor data transfer instructions of ARM processor. (8)

ii) Explain the ARM floating-point architecture. (8)

———————



Reg. No. :

Question Paper Code : 71757

B.E./B.Tech. DEGREE EXAMINATION, APRIL/MAY 2017

Seventh Semester

Electrical and Electronics Engineering

EE 6008 — MICROCONTROLLER BASED SYSTEM DESIGN

(Common to Electronics and Instrumentation Engineering, Instrumentation and Control Engineering)

(Regulations 2013)

Time : Three hours

Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. What are the benefits of having RISC architecture?

2. Define Brown out reset mode.

3. Mention the interrupts available in FTC Microcontroller.

4. Write an ALP to initialize the PORT A using PIC microcontroller.

5. Microcontroller based control is more advantageous than conventional control — Justify.

6. How is temperature sensor is interfaced with PIC Microcontroller?

7. Define Context Switching.

8. State the function of ARMulator and define its operations at various levels of accuracy.

9. Draw the structure of multicycle instruction of three stage pipeline operation.

10. What is the role of a co-pressure?

PART B — (5 × 16 = 80 marks)

11. (a) With neat functional block diagram explain the architecture of PIC16C7X Microcontroller in detail.

Or

(b) (i) Discuss in detail about memory organization of a PIC microcontroller. (8)

(ii) Explain the various addressing modes in PIC, for accessing data memory. (8)

12. (a) Explain the process and procedure to display constant strings and variable strings.

Or

(b) Explain the concept of interrupt logic and interrupt structure of PIC microcontroller with an example.

13. (a) Illustrate with suitable example how I^2 C communication is carried out in PIC Microcontroller.

Or

(b) Explain the operation of ADC interfacing with PIC Microcontroller.

14. (a) (i) Explain the Arm Programmer's Model in detail, with supporting diagram. (10)

(ii) Write the subroutine program to output a text string following a CALL Instruction using ARM processor. (6)

Or

(b) Write short notes on ARM MMU architecture.

15. (a) Elaborate the working principle of VLSI ISDN subscriber processor in detail.

Or

(b) Write short notes on

(i) 5 stage pipeline ARM organization.

(ii) Coprocessor dáta and register transfer.

———————