Set 2 Solution

Q.1. code - 37934

(01)

SEM V EXTC Microcontrollers & Applications

3 Hours

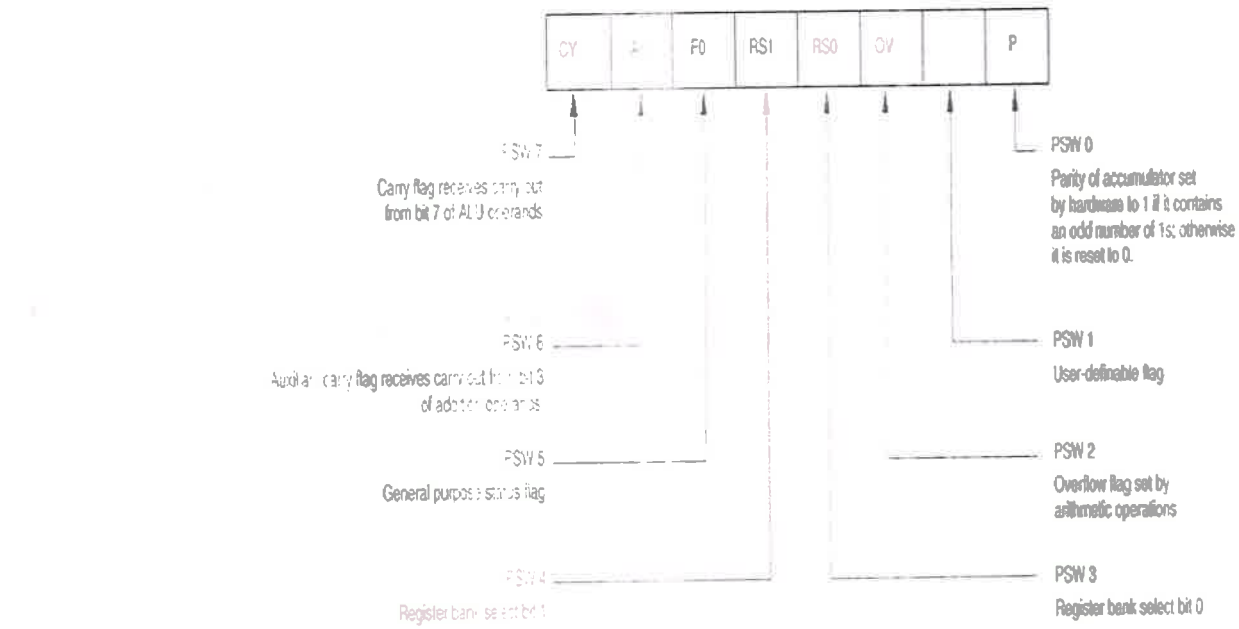
Total marks: 80

- Question no. 1 is compulsory
- Attempt any Three questions from remaining

Q1 a Explain Program Status word Register of 8051 Microcontroller 4

Marking Scheme: 2 Marks for labeled diagram & 2 Marks for bit explanation

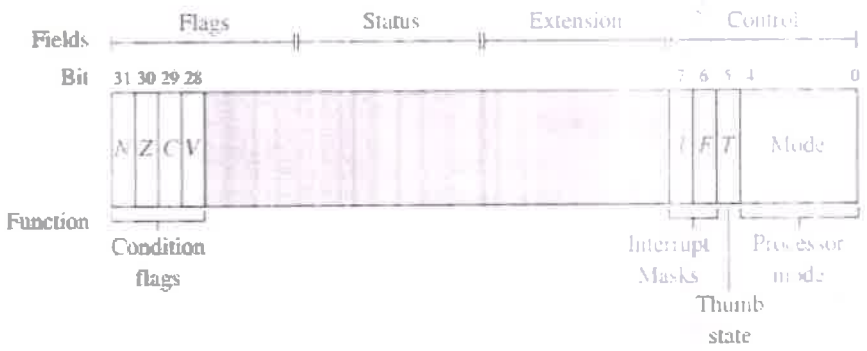
PSW register stores the important status conditions of the microcontroller. It also stores the bank select bits (RS1 & RS0) for register bank selection.



b Explain Current Program Status Register of ARM7. 4

Marking Scheme: 2 Marks for labeled diagram & 2 Marks for bit explanation

The cpsr is divided into four fields, each 8 bits wide: flags, status, extension, and control.In current designs the extension and status fields are reserved for future use. The control field contains the processor mode, state, and interrupt mask bits. The flags field contains the condition flags.Some ARM processor cores have extra bits allocated. For example, the J bit, which can be found in the flags field, is only available on Jazelle-enabled processors, which execute 8-bit instructions.



Field	Use
N	Negative flag, records bit 31 of the result of flag-setting operations.
Z	Zero flag, records if the result of a flag-setting operation is zero.
C	Carry flag, records unsigned overflow for addition, not-borrow for subtraction, and is also used by the shifting circuit. See Table A.3.
V	Overflow flag, records signed overflows for flag-setting operations.
Q	Saturation flag. Certain operations set this flag on saturation. See for example QADD in Appendix A (ARMv5E and above).
J	$J = 1$ indicates Java execution (must have $T = 0$). Use the BXJ instruction to change this bit (ARMv5J and above).
Res	These bits are reserved for future expansion. Software should preserve the values in these bits.
GE[3:0]	The SIMD greater-or-equal flags. See SADD in Appendix A (ARMv6).
E	Controls the data endianness. See SETEND in Appendix A (ARMv6).
A	$A = 1$ disables imprecise data aborts (ARMv6).
I	$I = 1$ disables IRQ interrupts.
F	$F = 1$ disables FIQ interrupts.
T	$T = 1$ indicates Thumb state. $T = 0$ indicates ARM state. Use the BX or BLX instructions to change this bit (ARMv4T and above).

c Explain Assembler Directives in 8051 microcontroller 4

Marking Scheme:4M for explanation of four Assembler Directives

DB, ORG, END, EQU

d Explain Features of ARM7 4

Marking Scheme:4M for correct explanation

Data size and instruction set

ARM processor is a 32-bit architecture

Most ARM's implement two instructions sets

03

32-bit ARM instruction set
16-bit Thumb instruction set
Data types
ARM instructions are all 32-bit words, word aligned thumb instructions are half-words
Internally all ARM operations are on 32-bit operands. When a byte loaded from memory, it is zero or sign-extended to 32-bits.
Processor Modes
ARM has seven basic operating modes
Mode changes by software control or external interrupts
Similar to RISC architecture (not purely RISC)
Variable cycle instructions (LD/STR multiple)
Inline barrel shifter
Auto-increment/decrement addressing modes

e Explain concept of Cortex-A, the Cortex-R and the Cortex-M. 4
Marking Scheme:4M for correct explanation

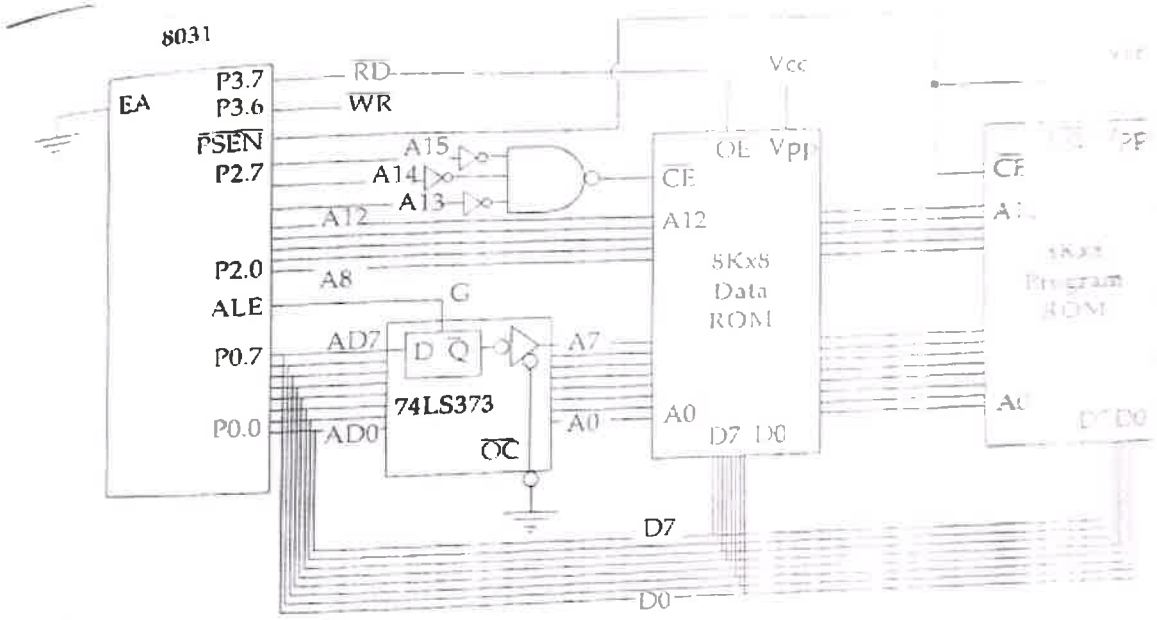
A-profile: Application
Cortex-A, for Application, is probably connected to a large amount of memory and runs at a relatively high clock speed.
Capable of running operating systems such as WinRT, Linux.
It can be used as the primary processor on mobile devices that require fast computational power, while using little power.
Applications: mobile phones, tablets, digital cameras, and just about any consumer mobile device.
Includes Cortex A-5,A7,A8,A9 A12,A15 and A17 core.
R-profile: Real-time
Cortex-R for real time applications.
Target high-performance real-time applications such as automotive systems, storage devices, medical instruments, digital TVs etc.
Includes Cortex- R4,R5 and R7.
M-profile: Microcontroller
ARM Cortex-M cores are intented for microcontroller use and consist of Cortex-M0, M0+, M1, M3 and M4.
It is often used to control hardware devices or to be an interface between hardware and another processor.
These are used in automotives, gaming consoles, Bluetooth and ZigBee,touchscreen and power management.

Q2 a Explain the Memory Interfacing of 8051 with 8K*8 Data ROM & 8K*8 PROM 10
Marking Scheme
2 Marks for calculating address lines: 13 bit address lines. A0-A12
2 Marks for showing 8051 signals in diagram

04

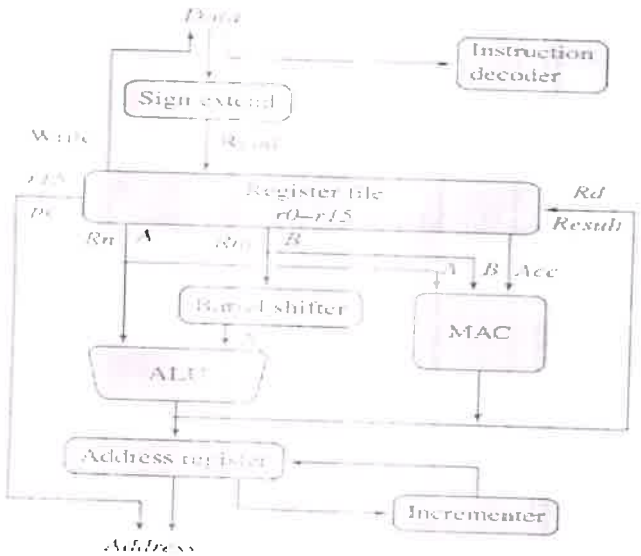
2 marks for showing decoder logic in diagram
4 marks for showing memory chip connections with 8051 microcontroller

For ROM, PSEN is used to activate both OE and CE.This is one of the most widely used address decoders74LS373:The 3 inputs A, B, and C generate 8 active low outputs Y0 – Y7 Each Y output is connected to CS of a memory chip, allowing control of 8 memory blocks by a single 74LS138 In the 74LS138, where A, B, and C select which output is activated, there are three additional inputs, G2A, G2B, and G1 G2A and G2B are both active low, and G1 is active high any one of the inputs G1, G2A, or G2B is not connected to an address signal, they must be activated permanently either by Vcc or ground,



b Draw & Explain dataflow model of ARM7 10

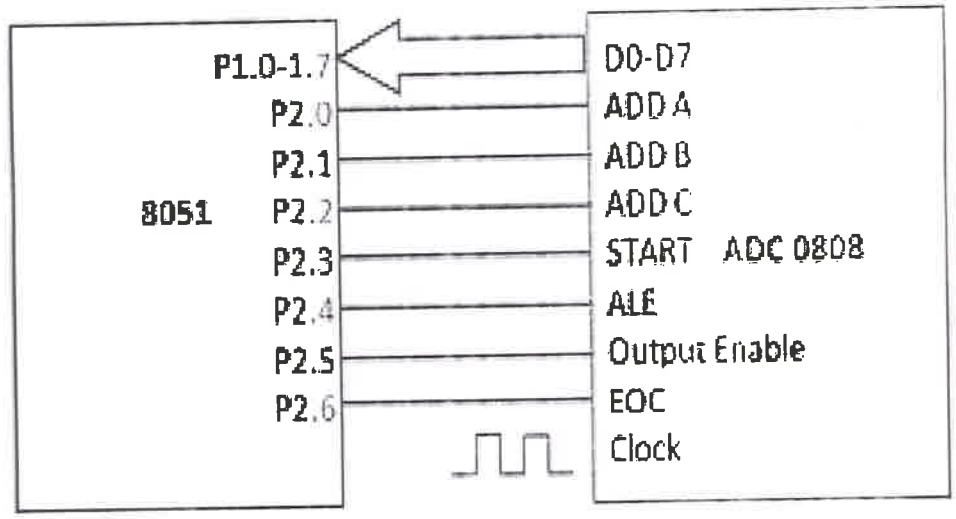
Marking Scheme:4M labeled for diagram & 6M for correct explanation



The ARM processor like RISC processor uses a load store architecture. This means it has two instruction types for transferring data in and out of the processor: load instructions copy data from memory to registers in the core and conversely the store instructions copy data from registers to memory. Data items are placed in the register file- a storage bank made up of 32 bit registers. Since the ARM core is a 32 bit processor, most instructions treat the registers as holding signed or unsigned 32 bit values. The sign extended hardware converts signed 8 bit and 16 bit numbers to 32 bit values as they read from memory and placed in register. ARM has two source registers Rn and Rm and single destination register, Rd. The ALU and MAC (Multiply accumulate unit) takes the registers values Rn and Rm from the A and B buses and compute result. One important feature of thr ARM is that register Rm alternatively can be preprocessed in the barrel shifter before it enters the ALU. After passing through the functional units the result in Rd is written back to the register file using the result bus. For load and store instructions the incrementer updates the address register before the core reads or writes the next register value from or to the next sequential memory location.

Q3 a Interface ADC 0808 with 8051 microcontroller. Write Assembly language Program to convert analog signal which is available on channel No 6 — 10

Marking Scheme: 3 Marks for working of ADC0808, 3 Marks for ADC0808 connection diagram with 8051 microcontroller along with explanation, 4 Marks for program along with algorithm or comments written in program



Sample Algorithm & Program
Start.
Select the channel.
A Low – High transition on ALE to latch in the address.
A Low – High transition on SC to reset the ADC
Wait for End of conversion (EOC) pin to become LOW If conversion is over.
Make Output Enable pin Low.
Take Data from the ADC's output
Make Output Enable pin High for next conversion
Stop

```
        MOV P1,#0FFh                    ; Making P1 as input port
        SETB P2.2                           ; SELECT CH 3
        SETB P2.1                       ; SELECT CH 3
        CLR P2.0                        ; SELECT CH 3
        CLR P2.4                         ; ALE LOW
        SETB P2.4                                   ; ALE HIGH
        SETB P2.6                       ; EOC HIGH
        Back:   CLR P2.3                ; SOC LOW
SETB P2.3                       ; SOC HIGH
HERE : JB P2.6, HERE            ; WAIT FOR EOC
CLR P2.5                    ; Enable Read or output enable
MOV A, P1
MOV 30h,A
SETB P2.5                       ; OE=1 FOR NEXT ROUND
SJMP Back
```

b Write a program for 8051 microcontroller to generate square waveform of 2kHz & 50% duty cycle at pin P1.5. Assume 8051 is operating at frequency 11.059MHz 10
Marking Scheme: 4 Marks calculation, 5 Marks for program, 1 Mark for comments or Algorithm
Sample Program

07

(a) T = 1 / f = 1 / 2 kHz = 500 us the period of square wave.
(b) 1 / 2 of it for the high and low portion of the pulse is 250 us.
(c) 250 us / 1.085 us = 230 and 65536 – 230 = 65306 which in hex is FF1AH.

```
 1 MOV TMOD,#01 ;Timer 0, 16-bitmode
 2 AGAIN: MOV TL1,#1AH ;TL1=1A, low byte of timer
 3 MOV TH1,#0FFH ;TH1=FF, the high byte
 4 SETB TR1 ;Start timer 1
 5 BACK: JNB TF1,BACK ;until timer rolls over
 6 CLR TR1 ;Stop the timer 1
 7 CLR P1.5
 8 CLR TF1 ;Clear timer 1 flag
 9 SJMP AGAIN ;Reload timer
10 END
11
```

OR

```
MOV TMOD,#01              ;timer 0, mode1, 16 bit mode
HERE: MOV TL0,#1AH        ; Load lower byte of timer
      MOV TH0,#0FFH       ; Load higher byte of timer
      CPL P2.1            ;toggle p1.0
      ACALL DELAY
      SJMP HERE           ;load TH and TL again
DELAY: SETB TR0           ; start timer 0
AGAIN: JNB TF0,AGAIN      ;monitor flag until
                          it rolls over
      CLR TR0             ;stop timer 0
      CLR TF0             ; clear timer 0 flag
      RET
```

10

Q 4 a Explain 8051 Timer operating modes

Marking Scheme: 3 Marks for explaining each mode in detail & 1M for diagrams

M1	M0	MODE
0	0	13-bit timer mode
0	1	16-bit timer mode
1	0	8-bit auto-reload mode
1	1	split mode

10

b Explain ARM interrupts along with Interrupt Vector Table.

Marking Scheme: 6 Marks for explaining in detail & 4M for diagrams

Exception(Interrupt), Vector Table:

In ARM there are two interrupt request levels in processor core---- IRQ and FIQ

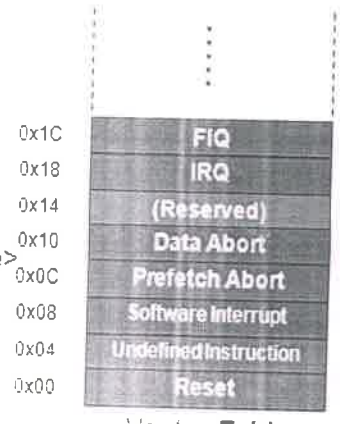
Two interrupt mask bits , 7 and 6 of CPSR (I and F which control masking of IRQ and FIQ respectively)

Exception	Shorthand	Priority	IV Address
Reset	RESET	1	0x00000000
Undefined instruction	UNDEF	6	0x00000004
Software interrupt	SWI	6	0x00000008
Prefetch Abort	PABT	5	0x0000000C
Data Abort	DABT	2	0x00000010
Interrupt	IRQ	4	0x00000018
Fast interrupt	FIQ	3	0x0000001C

09

- When an exception occurs, the ARM:
  - Copies CPSR into SPSR_<mode>
  - Sets appropriate CPSR bits
    - Change to ARM state
    - Change to exception mode
    - Disable interrupts (if appropriate)
  - Stores the return address in LR_<mode>
  - Sets PC to vector address
- To return, exception handler needs to:
  - Restore CPSR from SPSR_<mode>
  - Restore PC from LR_<mode>

This can only be done in ARM state.



Vector Table

Vector table can be at 0xFFFF0000 on ARM720T and on ARM9/10 family devices

Q 5 a Explain Addressing modes of ARM7 Processor with example in each. 10
Marking Scheme: 2 Marks each for explaining five addressing modes with example.

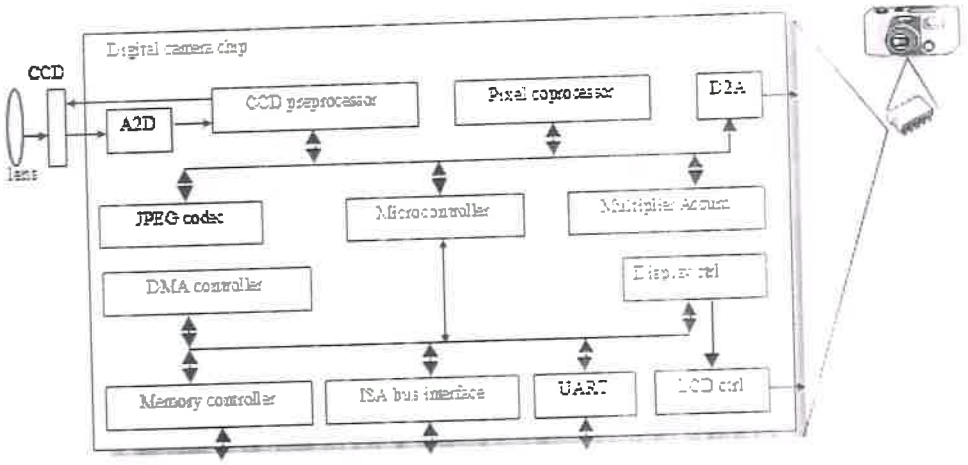
Name	Alternative Name	ARM Examples
Register to register	Register direct	MOV RO, R1
Absolute	Direct	LDR RO, MEM
Literal	Immediate	MOV RO, #15 ADD R1, R2, #12
Indexed, base	Register indirect	LDR RO, [R1]
Pre-indexed, base with displacement	Register indirect with offset	LDR RO, [R1, #4]
Pre-indexed, autoindexing	Register indirect pre-incrementing	LDR RO, [R1, #4]!
Post-indexing, autoindexed	Register indirect post-increment	LDR RO, [R1], #4
Double Reg indirect	Register indirect Register indexed	LDR RO, [R1, R2]
Double Reg indirect with scaling	Register indirect indexed with scaling	LDR RO, [R1, r2, LSL #2]

b Discuss Digital camera as an Embedded System 10
Marking Scheme: 4 Marks for correct diagram & 6M for explanation

- First, it performs a single function repeatedly. The system always acts as a digital camera, wherein it captures, compresses and stores frames, decompresses and displays

frames, and uploads frames.

- Second, it is tightly constrained. The system must be low cost since consumers must be able to afford such a camera. It must be small so that it fits within a standard-sized camera. It must be fast so that it can process numerous images in milliseconds. It must consume little power so that the camera's battery will last a long



Q 6 a Write short notes on(Any Two) 10

1. Internal Structure of PORT 1

Marking Scheme: 4 Marks for correct diagram & 6M for explanation 10

2. SCON in 8051

Marking Scheme: 4 Marks for correct diagram & 6M for explanation 10

3. Interrupts in 8051

Marking Scheme: 4 Marks for correct diagram & 6M for explanation