Scoring Indicators

Code : TED(IS) 5041 Embedded System Version: A

Qn. No.	Scoring Indicators	Split score	Total score
I	**Part A**		
3.	Port C pin 5 as input :- DDRC = DDRC & 0b11011111 OR DDRC = DDRC & ~(1<<5) (Assembly Language → CBI PORTC, 5)	2	2
1.	SFR → Special Function Registers Each register is dedicated to ~~specific~~ specific function. (I/O memory) Eg:- status, timer, I/O, ADC...	2	2
2.	Instruction Pipelining :- - Allows CPU to fetch and execute at same time. - Processor speed increases Previous instruction executes while next instruction fetches.	*Any 1	2
4.	Advantages of Raspberry Pi → Open source softwares → Internet wing Ethernet → Mini computer → More speed → More powerful.	2	2
5.	Embedded system Combination of hardware and software, either fixed or programmable, designed for a specific function.	2	2
II	**Part B**		
1.	Features of AVR family - 8 bit RISC single chip - Harvard architecture - ROM → 1K to 256K, Flash memory - Maximum 64K RAM - 3 to 86 pins for I/O	1×6	6

- 10 bit ADC, max 16 channels
- upto 6 timers, watch dog timer
- USART, I²C, SPI, CANbus

2. SRAM + EEPROM — 3×2 — 6

SRAM losses data when power is off.

So EEPROM is used to store data that should rarely be changed → options and settings.
SRAM for storing data and parameters that are changed frequently.

3. Structure of Assembly Language — 6 — 6

[Label:] mnemonic [operands] [; comment]

[] → optional
mnemonic → instruction
comment → not executed.

4. Macros — 6 — 6

→ Group of instructions used repeatedly for a task

→ .MACRO name → begin
... → body
.ENDMACRO → end

→ Upto ten parameters @0 to @9
→ Invoked by name.

5. Data types in C — 1×6 — 6

Unsigned char	8 bit	0 to 255 → widely used
char	8 bit	-128 to +127 (default)
unsigned int	16 bit	
int	16 bit	
unsigned long	32 bit	
long	32	
float	32	
double	32	

6. Interrupt priority

- Interrupt with highest priority served first
- Interrupt with lower address has higher priority.

	ROM
INT0	→ 0002
INT1	→ 0004
⋮	
Timer	
SPI	
USART	
ADC	
EEPROM	

↑ Priority

6 | 6

7. Categories of Embedded OS

- Non Real Time o/s
  - → Embedded NT → Single Board computer
  - → Windows XP → Use Win 32 applications and drivers
  - → Linux → Linux kernel, open source software
- Real Time → Within time constrain
  - → QNX Neutrino → ARM processor
  - → Vx Works
  - → Micro C/OS-II
  - → RT Linux
- Hand Held o/s
  - → Smart phones → Symbian, Windows CE

2×3 | 6

III UNIT I PART C

1. AVR family

Classic AVR :- Orginal AVR chip

Mega AVR :-
- 120 instructions
- ROM → 4k to 256k
- 28 - 100 pins IC
- Extensive Peripheral set

eg:- ATmega 8, ATmega 16 ..

Tiny AVR :-
- Limited instructions
- ROM 1k to 8k
- 8 to 28 pins
- Limited peripheral set

2×4 | 8

Eg:- ATtiny 13, ATtiny 25

Special Purpose AVR :- specific applications

Eg:- LCD controller (ATmega 169)

CAN controller (AT90CAN128) ...

2. Direct and Indirect

Direct - operand data in RAM, and address given in instruction — 3+4 — 7

- 0x0000 to 0xFFFF
- STS and LDC (IN and OUT)

Eg:- STS 0x520, R19 , LDS R19, 0x520

Register Indirect :-

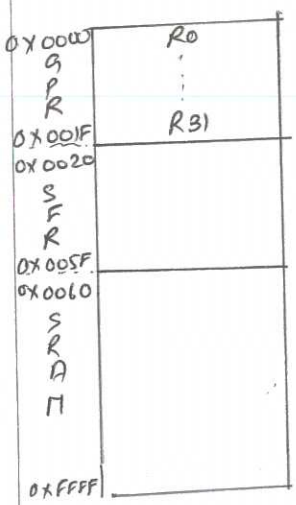
- operand data in RAM, Register X, Y or Z used as pointer to memory location
- LD and ST - 0x0000 to 0xFFFF
- X → R26 and R27, Y → R28 and R29, Z → R30 and R31

Eg:- ST X, R23 , LD R23, X

OR

IV B.

1. Data memory :- - store data — 3+3+2 — 8



- GPR, I/O and SRAM
- GPR → General purpose Registers
  - → 32 bytes, R0 to R31,
  - → 0x00 to 0xFF
- I/O (SFR)
  - → specific function
  - → Timers, ADC, Serial communication etc.
  - → at least 64 bytes
  - → Some AVR has extended I/O memory
- SRAM
  - → Data and parameters
  - → Size varies with chip.

2. Status Register :- — 1x7 — 7

D7	D6	D5	D4	D3	D2	D1	D0
I	T	H	S	V	N	Z	C

C → Carry, from D7

T → Bit copy storage

H → Half carry, carry from D3 to D4

S → Sign, 1 → Negative, 0 → Positive S = N ⊕ V

V → Overflow, Limit exceeds 0xFFFF (unsigned)

N → Negative, 1 → Negative, 0 → positive

Z → Zero, Result of an operation is zero

I → Interrupt, 1 → Interrupt enabled, 0 → disabled

UNIT II

V

1. I/O Port Programming

→ 32 pins, 4 ports with 8 bits — 2

→ PORTA, PORTB, PORTC and PORTD — 2

→ Three registers, PORTx, PINx and DDRx — 2

PORTx → To output data — 2

PINx → To input data

DDRx → To set direction, in or out

→ 1 → out, 0 → in

Eg:- DDRB = 0xFF → o/p for PORTB

DDRC = 0x00 → Port C for input

2. Logic Instructions

AND → AND Rd, Rr

OR → OR Rd, Rr

EXOR → EOR Rd, Rr

Complement → COM Rd (1's)

Negate → NEG Rd (2's)

Rd	Rr	Rd				
		AND	OR	EOR	COM	NEG
0	0	0	0	0	F	0
0	F	0	F	F	F	0
F	0	0	F	F	0	1
F	F	F	F	0	0	1

2 — 2 — 2 — 2 — 7

OR

VI

1. Conditional Branch Instruction

BREQ →	Branch if equal →	Z = 1
BRNE →	not equal →	Z = 0
BRSH →	Same or higher →	C = 0
BRLO →	Lower →	C = 1
BRLT →	Less than →	S = 1
BRGE →	Greater than →	S = 0
BRVS →	Overflow set →	V = 1
BRVC →	Overflow clear →	V = 0

1 × 8 — 8

2. Program :-

```
LDS   R19, 0X0240   → Load from 0X0240
LDS   R20, 0X0241   → Load from 0X0241
MUL   R19, R20      → Result in R0 and R1
STS   0X0242, R0    → Low byte
STS   0X0243, R1    → High byte
```

} 3 } 3 1 — 7

UNIT III

VII

1. Program :-

ASCII (~~Packed BCD~~ ASCII code) → To Unpacked BCD → Swap → Packed BCD

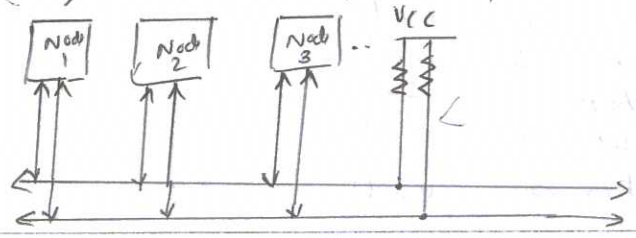
'4'	(34)	→	04	→	40 → 47
'7'	(37)		07		07

Logic + Program — 8

```
#include <avr/io.h>
void main(void)
{
    unsigned char x, y, data;   // declare three variables
    x = '4'; y = '7';           // input values
    x = x & 0F; y = y & 0F;     // masking upper nibble
    x = x << 4;                 // swapping
    data = x | y;               // combining
}
```

2. I²C Inter Integrated circuit (IIC) (TWI) — 7 — 7

→ serial communication
→ used in Sensors, RTC, EEPROM
→ short distance communication with acknowledgment
→ 2 pins SDA → Serial Data
SCL → serial clock
→ Pullup resistor for each bus line
→ Upto 128 devices (nodes)
→ Each node either master or slave
→ so, Master transmit, Master Receive, Slave transmit, slave receive, any one mode
→ Single byte or multi byte.
→ Registers → Data, Status, control, Address, Bit rate



VIII

OR

1. **Timer 0 Programming**

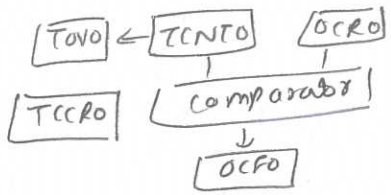
TCNT0 → Timer Counter Register, 8 bit, 0x00 to 0xFF — 8 — 8
→ starts counting from initial value loaded

TCCR0 → Timer / Counter Control Register

D7	D6	D5	D4	D3	D2	D1	D0
-	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00

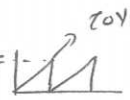
CS02-CS00 → clock Selector

0 0 0 → No clock (stop)
0 0 1 → clk
0 1 0 → clk/8
0 1 1 → clk/64
1 0 0 → clk/256
1 0 1 → clk/1024
1 1 0 → External clock at T0, (counter) falling edge
1 1 1 → External clock at T0 Rising edge

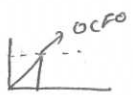


WGM 01,00 → Mode select
0 0 → Normal
0 1 → CTC
1 0 → PWM, Phase control
1 1 → Fast PWM

When TCNT0 exceeds 0xFF, TOV0 will set,



When TCNT0 = OCR0, OCF0 will set,



2. **Memory allocation in C**

SRAM → 64K 0x0000 to 0xFFFF
→ datas are stored in SRAM — 2 — 7

Flash → upto 4M 0x000000 to 0x1FFFFF
→ Programs are stored.
→ Program Counter indicate location — 2

EEPROM → Datas that should not be lost when power is OFF — 3
→ Using EEPRom data, address and control register
EEDR → data, EEAR → address, EECR → control, write / read

UNIT IV

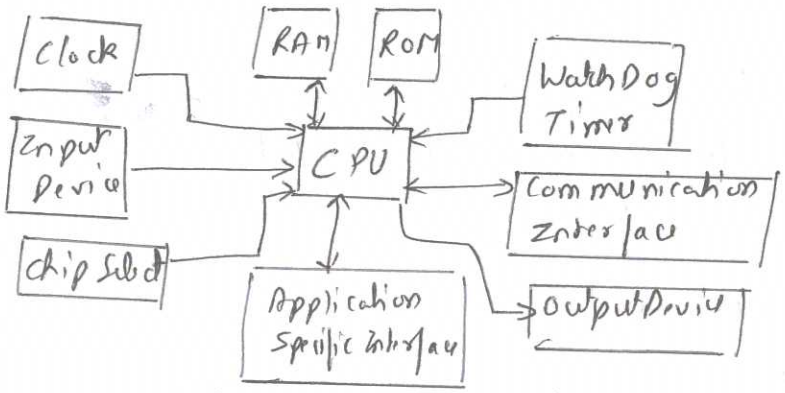
IX

2. **characteristic features of Embedded System**

- Task Specific
- Time Specific
- Minimal User Interface
- High Efficiency
- High Reliability
- Highly stable
- Low power
- Low cost

1 x 7 — 7

1. Architecture of Embedded system



4+4 (Fig)(Exp) | 8

CPU → Central Processing Unit
Memory → RAM and ROM
Input Devices + Output devices
Communication → Serial, Wireless, Bluetooth, Ethernet, ..
Application Specific → sensors, transducers, ...

OR.

X

1. Arduino Board :-
   - Circuit Board and Arduino IDE
   - Power section → USB or AC, regulator
   - Crystal oscillator for clock
   - Reset, Vcc, GND
   - Analog Pin A0 - A5
   - Digital Pin
   - Aref for ADC
   - TX and RX for serial data
   - Microcontroller, usually from ATMEL company.

1×8 | 8

2. Applications of ES
   → Consumer appliances → camera, DVD ..
   → Office appliances → fax, printer ...
   → Industrial applications → various factories
   → Medical applications → ECG, ESR
   → Communication → Mobile, PDA
   → Instrumentation → Analyzer
   → Security → Authentication & Verification
   → Finance → ATM

1×7 | 7