

Name : Kanishka K
 Degree & Programme : BTech CSE(CSBD)
 Course Code : INT 409.
 Examination : CIA I

Reg. No. : 125157030 - 20
 Year & Section : 1st Year - H
 Course Name : Embedded sec.
 Date : 16/8/24

16/8/24
 Signature of Invigilator

Kanishka
 Signature of Student

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
a																				
b																				
c																				
Total																				
Grand Total -																				

Begin Answering Here

Part-A

- 1) Programmed to do certain actions.
 2) 1 MHz - 300 MHz of speed.
 3) Does lightweight tasks
 4)
- #include <LPC21xx.h> ← Library
 Pnt main () { ← main function
 Pnt a=0; ← variable of type Pnt
 return a; ← returning variable.

3) $A = 0xAF2E$

= 1010111100101110
 ↑ 14th bit. ↑ 1st bit

to set 14th bit & reset 1st bit.

$A = (A | (1 \ll 14)) \& (~ (1 \ll 1))$

1010
 1111
 0010
 1110

Embedded - C

```
#include <LPC21XX.h>
```

```
int main () {
```

```
    int A = 0xAF2EFA;
```

```
A = ((A | (1 << 14)) & (~(1 << 2)));
```

```
return A;
```

SASTRA

Ans 4) $x = 0x12EFA$

on first operation it becomes

$0xEFA12$

which is a left circular shift of 8 bits

formula is

$$x = (x << 8) | (x >> 12)$$

generally

$$A = (A << x) | (A >> (N - x))$$

where x is the shift

on second operation it becomes

$0x2EFA0$

which is left shift

$$x = (x << 4)$$

5) $Q = 0xA00D$

$$Q = (Q << 8) | (Q >> 8)$$

for left circular

$$Q = (Q >> 8) | (Q << 8)$$

for right circular

6) Fine grained

1) Simple logic

2) used for minimum operation

3) has 1 bit logic

coarse grained

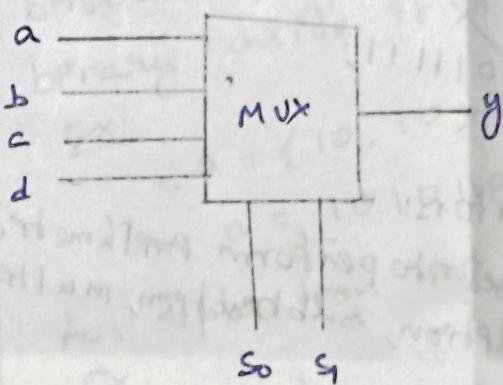
2) complex logic SASTRA

2) used on maximum operation.

3) has 4 bit look up tables (LUT).

2

2) 4:1 Mux



verilog.

```
module 4x1mux(a,b,c,d,s,y);
```

```
    input a,b,c,d;
```

```
    input output reg [1:0] s;
```

```
    output reg y;
```

```
    always @ (a,b,c,d,s):
```

```
        begin
```

```
            case(s):
```

```
                if 2'b00:
```

```
                    y = a;
```

```
                if 2'b01:
```

```
                    y = b;
```

```
                if 2'b10:
```

```
                    y = c;
```

```
                if 2'b11:
```

```
                    y = d;
```

```
                default
```

```
                    y = 0;
```

2

II) FPGAs are more

9) FPGAs are more energy efficient than micro controllers.

SASTRA

10) $K = \{b, a, 2\{3'b101\}, c\};$

$$b = 4'b1010$$

$$a = 4'b0001$$

$$c = 4'b1111$$

$$2\{3'b101\} = 101101$$

$$\underline{K = 101000011011011111};$$

Give ?

Part-B

12) a) Arithmetic operators:

These are used to perform Arithmetic operations like addition, subtraction, multiplication, etc.

Op - +, -, *, /, -.

Ex:

module add(a,b,c);

input a,b;

output c;

$$assign c = a+b;$$

end module.

here a & b are added & it is stored on c.

b) shifting operators:

These are used to shift the bits of an given variable.

Op - <<, >>, <<<, >>>.

Ex:

module ls(a, b);

 input a;

 output b;

 assign b = a << 2;

end module.

If $a = 0x01$, it becomes $0x04$.

c) Concatenation & replication operator.

To concatenate ~~any~~ any two binaries ' $\&$ ' is used. The variables or binary inside it is concatenated.

Ex:

$a = \{10, 10, 11, 01\}$

$\underline{a = 10101101}$

for replication put before ' $\&$ ' is used

Ex

$a = 3 \& 101$

$= 101010$

Verilog code

module concat(a, b, c);

 input a, b;

 output c;

 assign c = {a, b, 3'b03};

endmodule.

If $a = 1$, $b = 0$ then $c = \underline{10\ 000}$.

d) Relational operator:

These are used to compare ~~two~~ ~~more~~ ~~multiple~~ values

$>, <, \leq, \geq, ==, ==$

SASTRA

Ex:

```

module rel(a,b,c);
    input a,b;
    output c;
    begin(*)
        if (a > b) then
            assign c = a;
        else
            assign c = b;
        end
    endmodule

```

SASTRA

e) Logical operation:
These are used for logical operations

↑↑, ↑↑, ↑
and or not

ex - $(a < b \text{ } \& \text{ } b > c)$

will tell if b is greater than a length a

Encryption:

13 #include <LLPC 21xx.h>

unsigned char PL[] = { "THANJAVUR BIG
TEMPLE" };

unsigned int k = a;

unsigned char ct[~~18~~ + 18] // length 18

ptr p = 0;

for(i=0; i<18; i++) {

ct[i] = ((PL[i] - 0x41) + k) % 26 + 0x41;

}

Decryption:

~~Decryption~~
#include <LPC_21XX.h>
unsigned char CTF[?]

$$p_0 + K = q;$$

```
ptr k = 9;  
unsigned char p[18];
```

$\text{int } i = 0;$

```

int i=0;
for(i=0; i<18; i++) {
    cout[i] = ox41;
}

```

$$x = c + [i^2] - ox^4;$$

$$\text{if } f'(x - k) < 0 \text{ then } x$$

$$C[i] = \text{~~old value~~} C[i]$$

$$P[i] = \left[((C + [i] - 0 \times 41) - k) + 26 \right] \\ + 0 \times 41;$$

3

decryption logic

$$\text{Let's say } ct = c = 67 \text{ g}$$

but plain $t = z = 90$, $k = 3$

$$C = 67 - 65 = 2$$

$$c - k = -1;$$

$$\therefore \text{The C.E.} = 67 + 26 = 93$$

$$\therefore \text{The C.P.} = \\ \text{Now } 93 - 65 = 28$$

$$28 - k = 25$$

$$8 - k = \\ 25 + 65 = 90 = \underline{\underline{2}}$$

Our key = 9

Our key = 9
message = THANJAVUR BIG TEMPLE
with key

message = THANKS
to find the cipher message with key = 9

A	B	C	D	E	F	G	H	I	J	K	L
1	2	3	4	5	6	7	8	9	10	11	12
M	N	O	P	Q	R	S	T	U	V		
13	14	15	16	17	18	19	20	21	22		

$$W = X = Y = Z \quad (20 + 9 = 29 + 6 = 3)$$

$$T + 9 = C \quad (1 + 9 = 10)$$

$$U + 9 = J \quad (14 + 9 = 23)$$

$$N + 9 = W \quad (14 + 9 = 19)$$

$$J + 9 = S \quad (10 + 9 = 19)$$

$$A + 9 = I \quad (22 + 9 = 31 + 26 = 5)$$

$$V + 9 = E \quad (22 + 9 = 27 + 26 = 1)$$

$$U + 9 = D \quad (18 + 9 = 27 + 26 = 1)$$

$$R + 9 = A \quad (18 + 9 = 27 + 26 = 1)$$

$$B + 9 = K \quad (18 + 9 = 27 + 26 = 1)$$

$$I + 9 = R \quad (18 + 9 = 27 + 26 = 1)$$

$$G + 9 = P \quad (18 + 9 = 27 + 26 = 1)$$

$$T + 9 = C \quad (18 + 9 = 27 + 26 = 1)$$

$$E + 9 = N \quad (18 + 9 = 27 + 26 = 1)$$

$$M + 9 = V \quad (18 + 9 = 27 + 26 = 1)$$

$$P + 9 = Y \quad (18 + 9 = 27 + 26 = 1)$$

$$L + 9 = U \quad (18 + 9 = 27 + 26 = 1)$$

$$E + 9 = N \quad (18 + 9 = 27 + 26 = 1)$$

$$\therefore \text{the encoded message is}$$

~~W S J E D A K R P C N Y Y U N~~

to decode we must do the opposite process

$$C - 9 = T \quad (3 - 9 = -6 + 26 = 20)$$

$$Q - 9 = H \quad (17 - 9 = 8)$$

$$J - 9 = A \quad (10 - 9 = 1)$$

$$W - 9 = N \quad (23 - 9 = 14)$$

$$S - 9 = J \quad (19 - 9 = 10)$$

$$J - 9 = A \quad (10 - 9 = 1)$$

$$E - 9 = V \quad (5 - 9 = -4 + 26 = 22)$$

$$D - 9 = U \quad (4 - 9 = -5 + 26 = 21)$$

$$A - 9 = R \quad (1 - 9 = -8 + 26 = 18)$$

$$K - 9 = B \quad (11 - 9 = 2)$$

$$R - 9 = I \quad (18 - 9 = 9)$$

Kerns

Signature of the Student

DR

Initials of the Invigilator

$$P - q = G \quad (16 - 9 = 7)$$

$$C - q = T$$

$$N - q = E$$

$$V - q = M$$

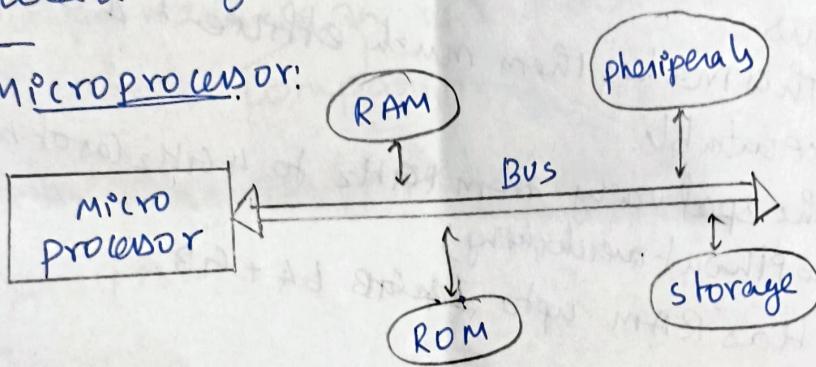
$$Y - q = P$$

$$U - q = L$$

$$N - q = E$$

Decoded msg = THANJAVUR BIG TEMPLE;

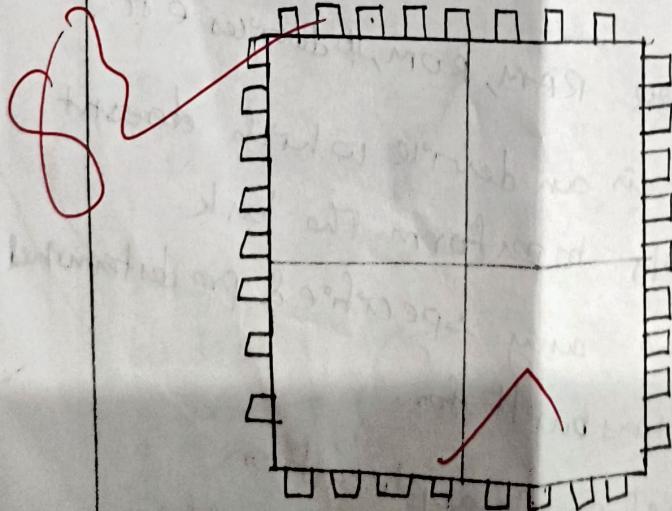
II) Microprocessor:



- The devices are connected to the microprocessor via a Bus.
- These devices are RAM, ROM, I/O devices etc.
- A micro processor is a device which doesn't have multiple units to perform the task.
- It doesn't have any specified predetermined task, which it was built for.
- It works based on the application.
- They are square in shape since they have a lot of pins for heat dissipation.

- SASTRA
- These run on high level programming language like C, C++.
 - These are designed to run complex codes & produce a throughput.
 - These are built with a idea in mind for it to solve any given task rather than clinging to a predefined set of application.
 - These devices are not inside the microprocessor there are somewhere on the chip/motherboard which are connected via a universal bus.
 - This makes them much efficient & easily executable.
 - The speed vary from 1 GHz to 4 GHz (as of now) (no P without overclocking)
 - Has RAM upto ~~128~~ GB 64 + GB.

FPGA:



- FPGAs are a type of microcontroller.
- This has 32 pins each having its own functions.



Kamal

Signature of the Student

H

Initials of the invigilator

- These FPGAs are designed for certain predefined task.
- They have an embedded chip, which can be programmed.
- The chip can be programmed to the user's need. & It will perform only that.
- The FPGAs chips are versatile & repeatable.
- They produce an high throughput & are faster comparatively.
- As they only perform a set of operations.

7) Continuous assignment:

It is a process of assigning a value continuously using functions like always @posedge

Procedural assignment

It is assigning the value procedurally using \leftarrow statement used on shift regs