

SEAT NO: A13A04F

ROLL NO: 47

NAME: YASH
SARANG

SEMESTER: IIIRD

SUBJECT: DLCOA

PAGE NO: 1/17

Sakshiyaash

Q.

- 1) Option d: All of the above
- 2) Option B: (36.246)₈
- 3) Option C: $Q_0 = 1$.
- 4) Option C: Arithmetic right shift
of A, Q and Q-1.
- 5) Option D: 120, 180.
- 6) Option D: A 10 KHz square wave.
- 7) Option A: State Table method.
- 8) Option B: Vertical organization.
- 9) Option C: A set associative cache
has lower number of conflict
misses than a direct mapped
cache of same size.
- 10) Option C: Pipelining reduce the latency
of each individual instruction.

Pg no. 3/17

Santaygash

Q. 2. A)

①
→

$$(340)_{10} = (?)_{XS3}$$

$$\begin{array}{r}
 3 \quad 4 \quad 0 \\
 +3 \quad +3 \quad +3 \\
 =6 \quad =7 \quad =3
 \end{array}$$

0110 0111 0011

$$\therefore (340)_{10} = (0110 \ 0111 \ 0011)_{XS3}$$

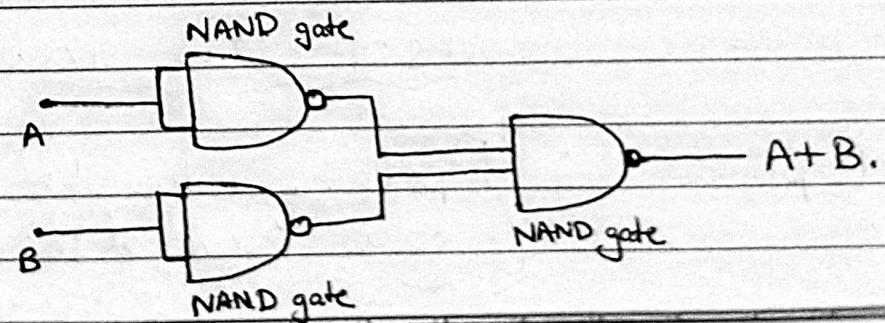
ii
→

$$(DADA)_{16} = (?)_{10}$$

$$\begin{aligned}
 (DADA)_{16} &= (13 \times 16^3) + (10 \times 16^2) \\
 &\quad + (13 \times 16^1) + (10 \times 16^0).
 \end{aligned}$$

$$= (56026)_{10}$$

$$\therefore (DADA)_{16} = (56026)_{10}.$$

iii
→

Pg no. 4/17

~~Sprangash~~

Q 2.

A)

iv

$$(3A9D.AOC)_{16} = (\quad)_2$$

$$= 3 \ A \ 9 \ D \ . \ A \ O \ C$$

$$= 0 \ 0011 \ 1010 \ 1001 \ 1101.1010 \ 0000 \ 1100$$

$$\therefore (3A9D.AOC)_{16} = (0 \ 0011 \ 1010 \ 1001 \ 1101.1010 \ 0000 \ 11)_2$$

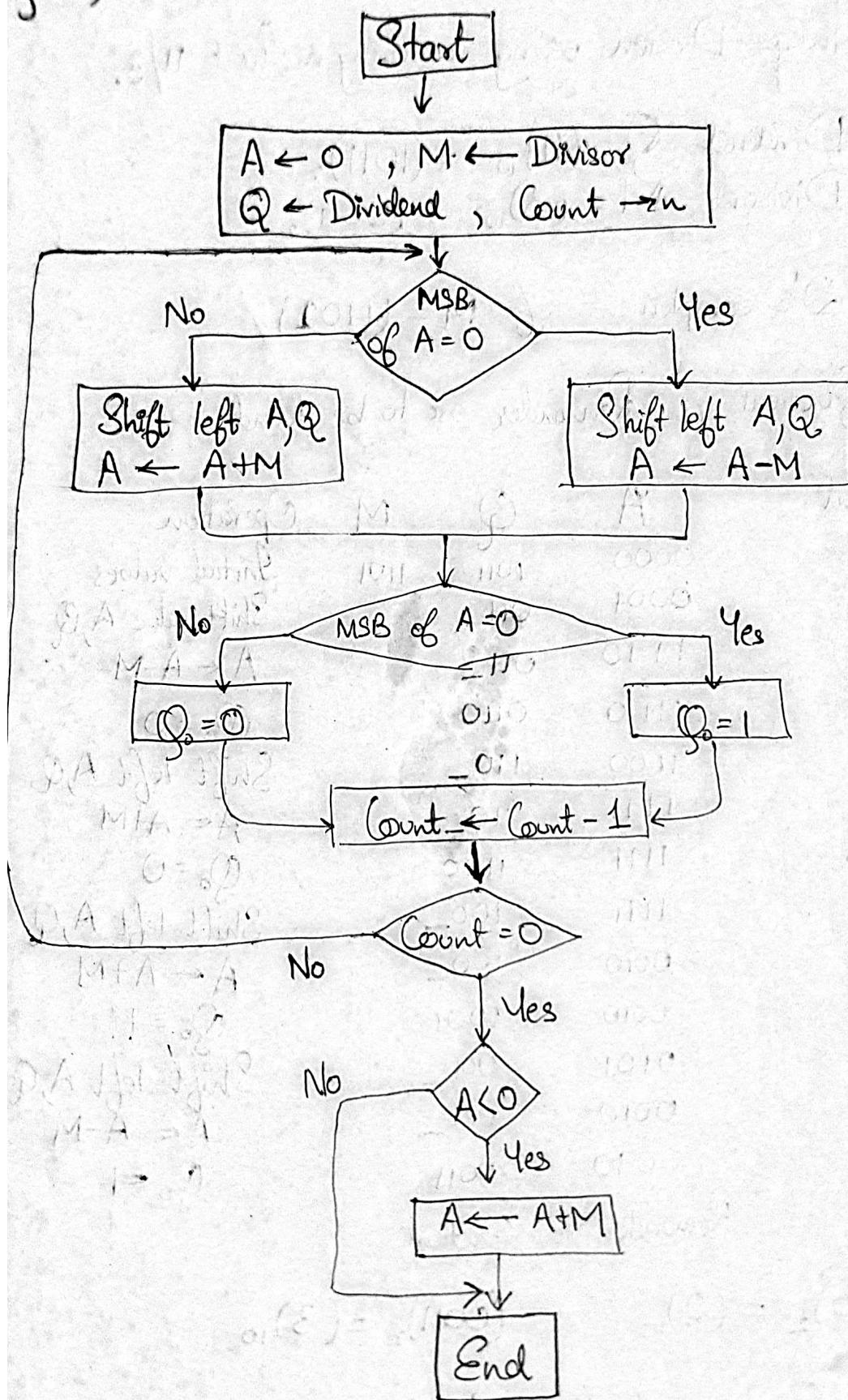
v

$$\rightarrow (52)_{10} = (110100)_2$$

Gray code :	1^{st}	\rightarrow	1
	2^{nd}	\rightarrow	$1+1 = 0$
	3^{rd}	\rightarrow	$0+1 = 1$
	4^{th}	\rightarrow	$1+0 = 1$
	5^{th}	\rightarrow	$0+1 = 1$
	6^{th}	\rightarrow	$0+0 = 0.$

$\therefore (52)_{10}$ to gray code is $(10 \ 1110)$.

(Q. 2.B)



Q2.B)

Integer Division using restoring method 11/3.

$$\text{Dividend } Q = (11)_{10} = (1011)_2$$

$$\text{Divisor } M = (3)_{10} = (0011)_2$$

Q's complement of M $\hat{=} (1101)$

Quotient & Remainder are to be found.

Count	A	Q	M	Operation
4	0000	1011	1101	Initial values
	0001	011_		Shift value A, Q
	1110	011_		$A \leftarrow A - M$
3	1110	0110		$Q_0 = 0$
	1100	110_		Shift left A, Q
	1111	110_		$A \leftarrow A + M$
2	1111	1100		$Q_0 = 0$
	1111	100_		Shift left A, Q
	0010	100_		$A \leftarrow A + M$
1	0010	1001		$Q_0 = 1$
	0101	001_		Shift left A, Q
	0010	001_		$A \leftarrow A - M$
0	0010	0011		$Q_0 = 1$
		Quotient		
		Remainder		

$$(0010)_2 = (2)_{10} \quad (0011)_2 = (3)_{10}$$

$$\therefore \text{Quotient} = (3)_{10} \quad \& \quad \text{Remainder} = (2)_{10}$$

Q3. A) 39887.5625 to IEEE 64-bit
 \rightarrow

Step 1] Convert decimal into binary.

$$(39887.5625)_{10} = (100110111001111.1001)_2$$

Step 2] Normalization.

$$(-1)^s \times 1.M \times 2^{TE}$$

$$(-1)^0 \times 1.00110111001111001 \times 2^{15}$$

Step 3] Calculate the bias exponent.

$$B.E = T.E + \text{bias}$$

$$= 15 + 1023.$$

$$B.E = (1038)_{10}$$

Step 4) Convert the bias exponent value to binary.

$$B.E = (1038)_{10} = (10000001110)_2$$

Q3.A)

Step 5] Substitute in IEEE Double precision floating point.

0	1 0000001110	00110111001111001000000000000000 00000000000000000000000000000000
---	--------------	--

S	E	M
1 bit	11 bit	52 bit.

∴ The number 39887.5625 converted from decimal system (base 10) to 64 bit double precision IEEE 754 binary floating point is:

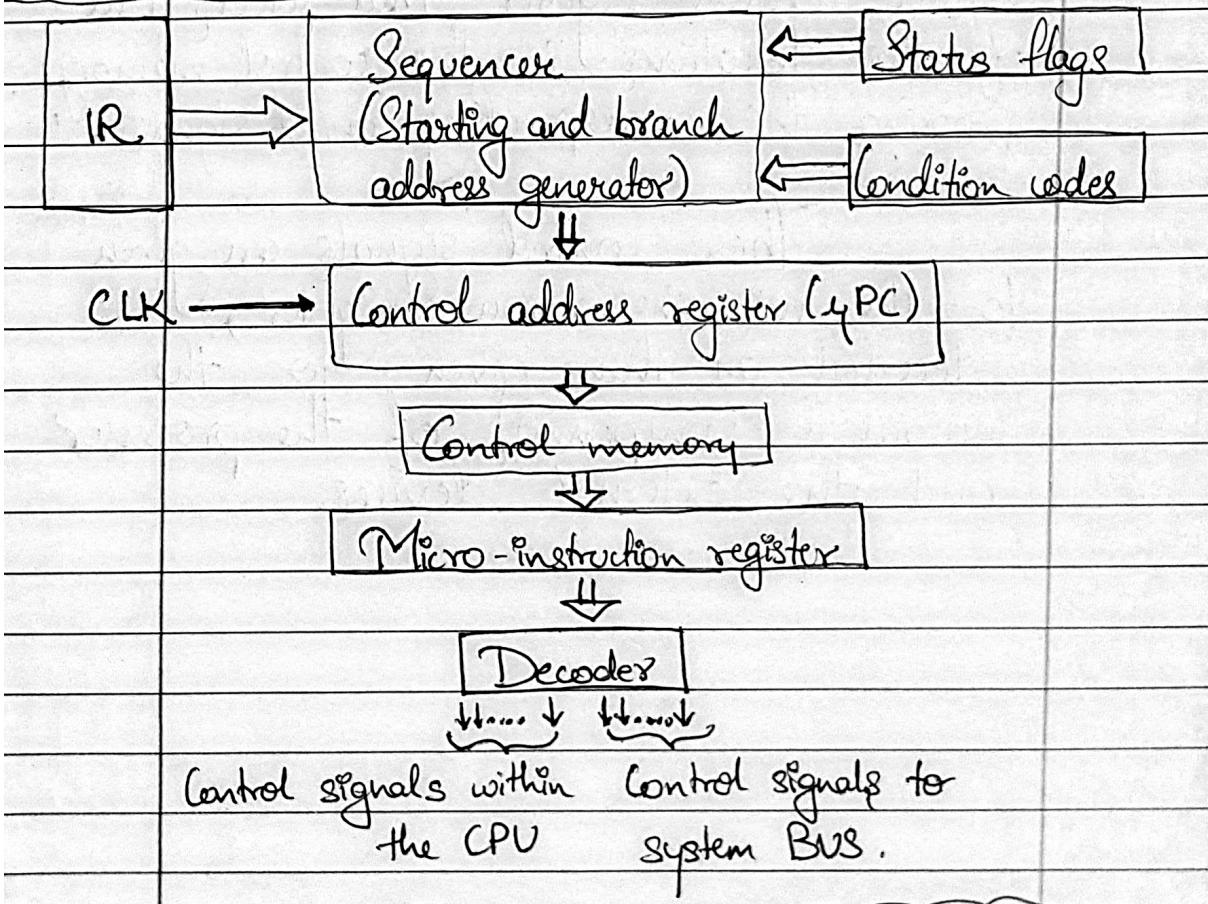
0-10000001110-0011 0111 1001 1111 0010 0000 0000 0000
0000 0000 0000 0000 0000.

Q 3. B)

→ Micro programmed control unit generates control signals based on the microinstructions stored in a special memory called as the control memory.

Each instruction points to a corresponding location in the control memory that loads the control signals in the control register.

The control register is then read by a sequencing logic that issues the control signals in a proper sequence.



Micro programmed control unit

Pg no. 10/17

Sarangshah

(Q 3. B)

The implementation of the microprogrammed control unit is shown in the diagram above.

The Instruction Register (IR), Status flag and condition codes are read by the sequencer that generates the address of the control memory location for the corresponding instruction in IR.

This address is stored in the Control address register that selects one of the locations in the control memory having the corresponding control signals.

These control signals are given to the microinstruction register, decoded and then given to the individual components of the processor and the external devices.

Pg no. 11 / 17

Sarkarjish

Q4. A)

Pipeline hazards are situations that prevent the next instruction in the instruction stream from executing during its designated clock cycles.

Any condition that causes a stall in the pipeline operations can be called a hazard.

Primarily there are 3 types of hazards:

i) Data Hazards

A data hazard is any condition in which either the source or the destination operands of an instruction are not available at ~~any~~ the time expected in pipeline.

As a result of which some operation has to be delayed and the pipeline stalls.

For example: $A = 3 + A$ and
 $B = 4 * A$.

In this sequence, the second instruction needs the value of 'A' computed in the first instruction.

Thus, the second instruction is said to be dependent on the first.

If the execution is done in a pipelined processor, it is highly likely that the interleaving of those two instructions can lead to incorrect results due to data dependency between the instructions.

Pg no 12/17

Gangapath

Q4.A)

ii) Structural hazards.

This situation arises mainly when two instructions require a given hardware resource at the same time and hence for one of the instruction, the pipeline needs to be stalled.

The most common case is when memory is accessed at the same time by two instructions. One instruction may need to access the memory as part of the Execute or write back phase while other instruction is being fetched.

In this case, if both the instructions and data reside in the same memory. Both the instructions can't proceed together and one of them needs to be stalled till the other is done with the memory access part. Thus, in general sufficient hardware resources are needed for avoiding structural hazards.

iii)

Control hazards:

The instruction fetch unit of the CPU is responsible for providing a stream of instructions to other the execution unit. The instructions fetched by the fetch unit are in consecutive

Pg no. 13/17

Saxengash

(Q4. A)

memory locations and they are executed.

However, the problem arises when one of the instructions is a branching instruction to some other memory location.

Thus all the instruction fetched in the pipeline from consecutive memory locations are invalid now and need to be removed.

(Also called the flushing of pipeline). This induces a stall till new instructions are again fetched from the memory address specified in the branch instruction.

Thus the time lost as a result of this is called a branch penalty.

Often dedicated hardware is incorporated in the fetch unit to identify branch instructions and compute branch addresses as soon as possible and reducing the resulting delay as a result.

Difference between delayed branch and branch prediction:

Delayed branch and branch prediction are two ways of mitigating the effects of a long execution pipeline. Without them, the pipeline needs to stall whenever a conditional branch is taken,

Pg no 14/17

Sarangshree

Q4.A)

Because the instruction fetch mechanism can't know which instruction should be executed next after the branch instruction until the computations on which it depends are complete.

Delayed branch simply means that some number of instructions that appear after the branches in the instruction stream will be executed regardless of which way the branch ultimately goes.

Branch prediction is more hardware-oriented approach, in which the instruction fetcher simply "guesses" which way the branch will go, executes instructions down the path, and if it later turns out to have guessed wrong, the results of those instructions are thrown away.

Pg no. 15/17

S. Ganguly

Q4.
c)

→ Interleaved memory.

The memory system in which successive addresses are evenly spread across memory bank to compensate for the relatively slow speed of DRAM.

The contiguous memory reads and writes are using each bank in turn, resulting in higher memory throughputs due to reduced waiting for memory banks to become ready for desired operations.

There are 2 types of Interleaving

i) Low order interleaving.

The lower order address lines eg. A0-A12, used to identify location in each bank whereas higher order address lines are decoded to generate chip select (CS) of individual memory chip. In this low order interleaving, successive addresses will be allocated in the same memory chip.

ii) High order interleaving.

In this case, higher order address

Pg no. 16/17

Sabangadi

Q. 4
c)

Lines i.e A3-A15 are used to identify location in each bank whereas lower order address lines A0,A1,A2 will be decoded to generate CS of individual chip. This will allocate successive addresses in the different memory banks.

Associative memory (CAM)

This memory is also known as content addressable memory. Its hardware organization consists of argument register to hold CPU's request in argument register. The result of CPU's request will be masked and remaining unmasked bits are then parallelly searched against each word in the associative memory array.

During this parallel search, as soon as the match is found the corresponding match register bit will be set and the data word from this matched location is set on the output for CPU.

The associative memory finds an application where search time is critical and very short. For example, The

Q4.c)

directory search in the computer or website search in the Internet system makes use of associative memory.

In this associative memory, if the matched words are contiguous they will be placed on Output, one after another without any intermediate gap. As a result, the data transfer rate will be very fast for associative memory.