Result (AQ) = 1110 1001

0001 0100 → 1's complement

+ 1

0001 0101 → 2's complement

Ans This is -21 in two's complement (which is correct since $-7 \times 3 = -21$).

→ Ex: -7×-3

-7 in 4-bit binary = 1001

-3 in 4-bit binary = 1101

Initialization:

- M = 1001 (-7)
- A = 0000
- Q = 1101 (-3)
- Q-1 = 0
- Count = 4

Steps:

	A	Q	Q-1	Count	Remark
	0000	1101	0	4	Initialize
*	~~1001~~				
+	0111				
	0111	1101	0		A - (-M)
	↓↘				
	0011	1110	1	3	Right shift
+	1001				
	1100	1110	1		A + (-M)
	↓↘				
	1110	0111	0	2	Right shift

Q.5] Multiply $7 \times -3, -7 \times 3, -7 \times -3$

→ Binary of 7 = 0111
Binary of 3 = 0011

-3 in 4-bit two's complement = 1101

$$\begin{array}{r} 0011 \\ \hline 1100 \\ +\quad\ 1 \\ \hline 1101 \end{array} \rightarrow [-3]$$

Multiplication :

- N = 0111 & -N = $\begin{array}{r} 0111 \\ \hline 1000 \\ +\quad\ 1 \\ \hline 1001 \end{array}$ → 2's complement
- A = 0000
- Q = 1101
- Q.1 = 0
- Count = 4

Assignment No-2

Q.1] Write short notes on: Programed I/O, Interrupt driven I/O, and DMA based I/O.

→ 1. Programmed I/O

i. In this method, the CPU is responsible for controlling the entire data transfer process.

ii. The CPU continously checks the device status to see if it is ready for data trasfer.

iii. Once the device is ready, the CPU reads/writes data to/from the device.

iv. Since the CPU is occupied with polling, it cannot execute other tasks, leading to inefficient CPU utilization.

v. Example: Reading a character from the keyboard using a loop to check if a key is pressed.

2. Interrupt - Driven I/O

i. Instead of continously polling, the CPU issues a command to the device and proceeds with other tasks.

ii. When the device is ready for data traveler, it sends an interrupt signal to the CPU.

A	Q	Q-1	Count	Remark
1110	0111	0		
+ 0111				
0101	0111	0		A - (-M)
↓				
0010	1011	1	1	Right shift
↓				
0001	0101	1	0	Right shift

Result (AQ) = 0001 0101

Ans This is 21 (which is correct since $-7 \times -3 = 21$)

Q.4] Perform the division of following number using Restoring and Non-restoring division algorithm. (17/5)

→ 1. Restoring Division Algorithm.

Initialize

- Dividend (N) = 17 (10001)
- Divisor (D) = 5 (00101)
- Remainder (A) = 00000
- Quotient Q = 10001 (dividend)
- n = 5 bits.

M = 00101

∴ -M = 11011

-M = 00101

11010

+ 1

11011

Q.3] What is the need of D.M.A in computer system? Explain in detail its operation in various modes.

→ i. Reduces CPU overhead: The CPU is freed from handling data transfers, allowing it to perform other task.

ii. Efficient Data transfer: DMA can transfer large blocks of data faster than CPU controlled methods.

iii. Improves System performance: By minimizing CPU intervention, system throughput process

iv. Support High-speed devices: Essential for peripherals like SSDs, GPUs, and network cards that need fast data movement.

→ Operation of DMA in various modes:

i. Burst Mode (Block transfer Mode)

- The DMA Controller transfer a large block of data in a single continuous operation.
- The CPU is completely halted during transfer.

Example: Copying a large file from SSD to RAM

→ N. Shift left 00111 00010

+ 11011

Sub m [1]00.0 00011

1's complement

Result :

- Quotient (Q) = 00011 → 3
- Remainder (A) = 00010 → 2

Hus This matches 17/5 = 3 (Quotient) / 2 Remainder.

Eg: -7×3

-7 in 4-bit binary = $\begin{array}{r} 0111 \\ 1000 \\ +\quad\ 1 \\ \hline 1001 \end{array}$

$\therefore -7$ in 4-bit binary = 1001

3 in 4-bit binary = 0011

→ Initializing

M = 1001 (-7) Q = 0011 (3)
A = 0000 Q-1 = 0 count = 4

Steps:

A	Q	Q-1	count	Remark
0000	0011	0	4	Initialize
+ 0111				
0111	0011	0		A - (-M)
↓ ↘				
0 011	1001	1	3	Right shift
↓ ↘				
0 0 01	1100	1	2	Right shift
+ 1 0 01				
1 0 1 0	1100 1			A + (-M)
↓ ↘				
1 101	01100		1	Right shift
↓ ↘				
1 1 1 0	10110		0	Right shift

iii. The CPU then pauses its current execution, services the interrupt and resumes normal operation.

iv. This method reduces CPU wastage but still requires CPU intervention for each data transfer.

Example: A printer sending an interrupt to the CPU when it finishes printing a page, allowing the CPU to send the next page.

3. DMA (Direct Memory Acess)

i. A dedicated hardware module called a DMA controller handles data transfer between memory and the I/O device.

ii. The CPU initializes the transfer by providing details (source, destination, size) and delegates control to the DMA controller.

iii. The DMA controller independently transfer the data and interrupts the CPU only when the transfer is complete.

iv. This method minimizes CPU overhead, making it highly efficient for high-speed data transfers.

Example: Transferring a file from a hard disk to RAM without CPU intervention.

Operations	A	Divided (Q)	Count Steps
Initial values	00000	10001	5
• Shift left	00001	0001_	
+	11011		
Sub m	11100	0001_	
		1's complement	
• Restore (Add M)	00001	0001 0	4
• Shift left	00010	0010_	
+	11011		
Sub m	11101	0010_	
Restore (Add m)	00010	00100	3
• Shift left	00100	0100_	2
+	11011		
Sub m	11111	0100_	
Restore (Add m)	00100	01000	
• Shift left	01000	1000_	1
+	11011		
Sub m	00011	10001	
• Shift left	00111	0001_	0
+	11011		
Sub m	00010	00011	

2. Cycle Stealing Mode

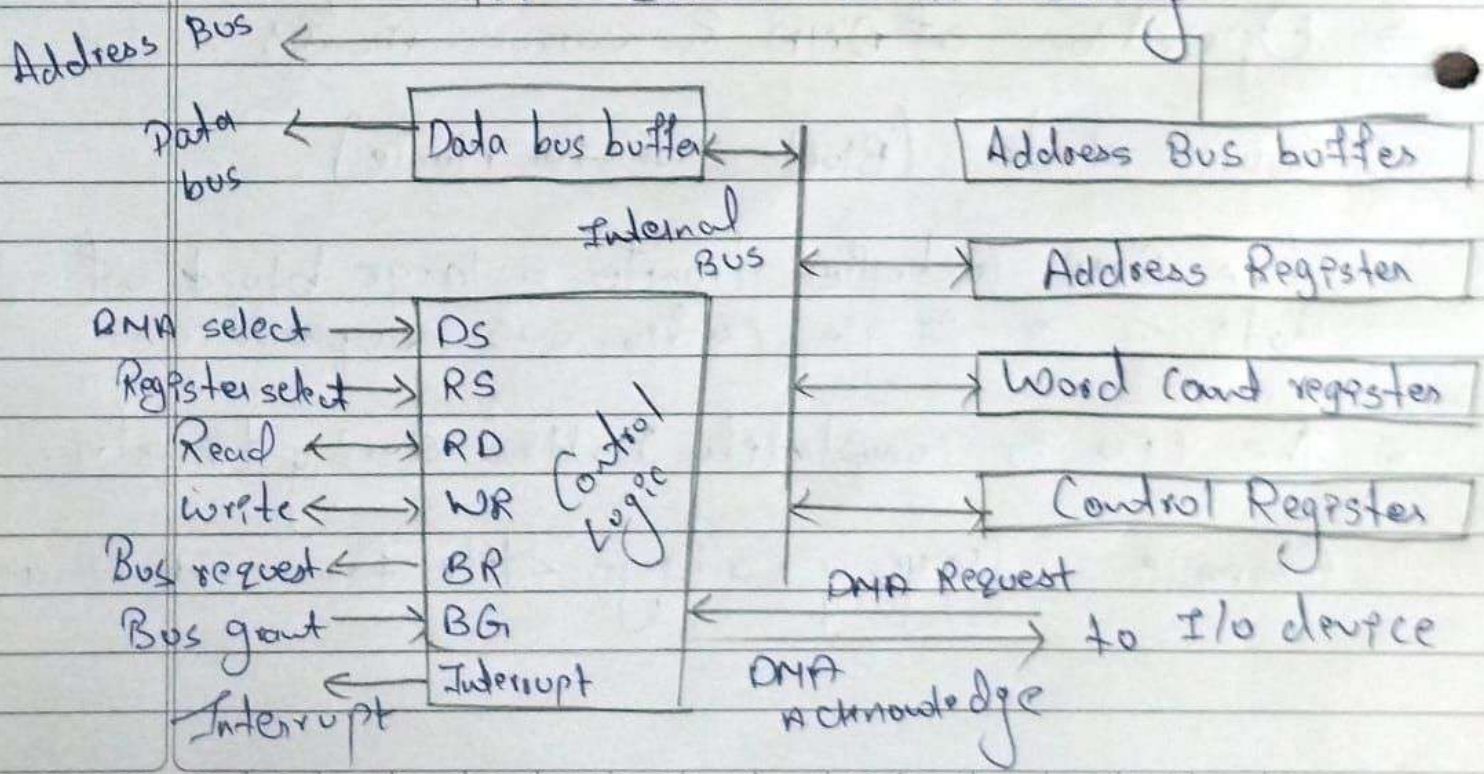
- The DMA controller transfer one byte or word of data at a time, temporarily pausing the CPU.
- The CPU and DMA takes turn accessing the memory preventing a complete halt.

Example : Data transfer from ADC to memory while the CPU is processing other tasks.

3. Transparent Mode

- The DMA controller transfers data only when the CPU is not using the system bus.

Example : Background data loading in memory for smooth multitasking.



* Working of DMP Process

1. CPU initiates transfer, it provides the DMA controller with source, destination addresses and transfer size.
2. The CPU is temporarily relieved, and DMAC handles the transfer.
3. Depending on the mode, data moves from I/O to memory or vice versa.
4. Once the transfer is complete, the DMA controller sends an interrupt to the CPU, signaling the end of operation.

Q.6] Explain DMA based data transfer techniques for I/O Devices.

-> Direct Memory Access (DMA) is a method where I/O devices transfer data directly to/from memory without CPU intervention, improving speed and efficiency.

Key Steps in DMA transfer

1. CPU Initialization

-> CPU configure the DMA controller (DMAC) with

Q.3] Represent following number in IEEE 754 format for Single & Double precession floating-point number representation
$(309.1875)_{10}$

→ Given number is = 309.1875
Sign bit = 0 (+ve)

i. Step 1: Converting the given number to binary format.

2	309	
2	154	1
2	77	0
2	38	1
2	19	0
2	9	1
2	4	1
2	2	0
	1	0
		1

$0.1875 \times 2 = 0.375 \rightarrow 0$
$0.375 \times 2 = 0.75 \rightarrow 0$
$0.75 \times 2 = 1.5 \rightarrow 1$
$0.5 \times 2 = 1 \rightarrow 1$

$309 = 100110101$ $0.1875 = 0011$

ii. Step 2: Normalising the binary number.

$[100110101.0011]_2$

$[1.00110101 0011 \times 2^8]$

Finding the value of E'

$E' = E + 127$

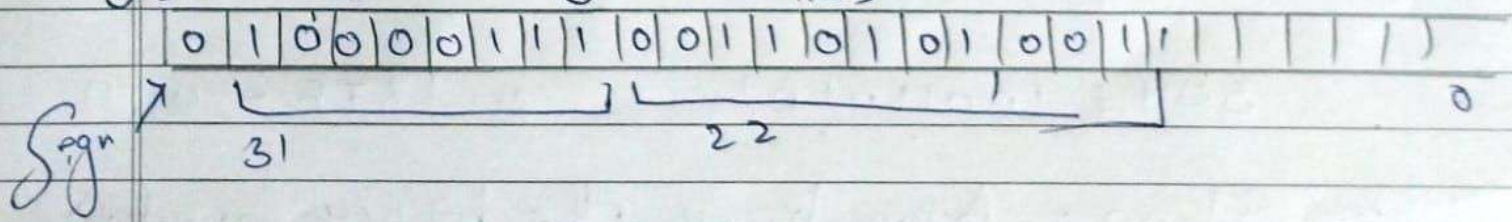
$= 8 + 127$

$E' = 135$

Converting the value of E' to binary

2	135	
2	67	1
2	33	1
2	16	1
2	8	0
2	4	0
2	2	0
	1	0
		1

iii) Step 3: Single Precission format (32-bit)

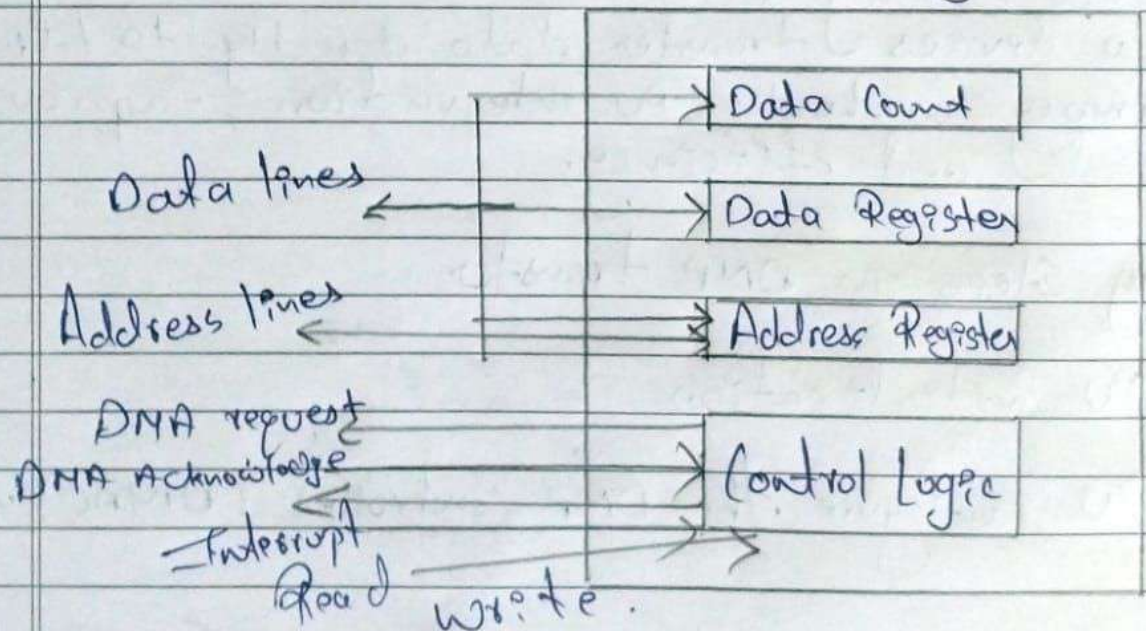


- Memory address
- I/O device address
- Data Size
- Transfer direction

2. DMA Request (DRQ)
- The I/O device signal the DMAC when ready to transfer data.

3. Bus Control Handover
- DMAC requests the system bus via HOLD signal.
- CPU releases the bus and acknowledges with HLDA (Hold Acknowledge)

4. Data Transfer
- DMAC takes over and moves data directly between I/O device and memory.



A	Q	Q-1	Count	Remark
0000	1101	0	4	Initialize
+ 1001				
1001	1101	0		A + (-M)
↓↘				
1100	1110	1	3	Right Shift
+ 0111				
0011	1110	1		A + M
↓↘				
0001	1111	0	2	Right Shift
+ 1001				
1010	1111	0		A + (-M)
↓↘				
1101	0111	1	1	Right Shift
↓↘				
1110	1011	1	0	Right Shift

Result (AQ) = 1110 1011

0001 0100 → 1's complement

\+ 1

0001 0101 → 2's complement

Ans This is -21 in two's compliment (which is correct since 7 x -3 = -21).

iv) Double Precission Format (64 bit)

finding $E' = E + 1023$

$= 8 + 1023$

$E' = 1031$

Converting E' value to binary

2	1031	
2	515	1
2	257	1
2	128	1
2	64	0
2	32	0
2	16	0
2	8	0
2	4	0
2	2	0
	1	0
		1

Sign → E' → Mantissa

0	1	0	0	0	0	0	0	0	1	1	1	0	0	1	1	0	1	1	0	1	0	1	0	0	1	1					

0

Result:

- Quotient (Q) = 00011 = 3
- Remainder (A) = 00010 = 2

This matches 17/5 = 3 (Quotient) / 2 (Remainder)

→ Non-Restoring Division Algorithm

Operations	A	Dividend (Q)	count
Initialization	00000	10001	5
→ i. Shift left	00001	0001_	4
"0" so sub m	+ 11011		
Sub m	11100	00010	
		(1's compliment)	
→ ii. Shift left	11000	0010_	3
"1" so add m	+ 00101		
Add m	11101	00100	
→ iii. Shift left	11010	0100_	2
	+ 00101		
Add m	11111	01000	
→ iv. Shift left	11110	1000_	1
	+ 00101		
Add m	00011	10001	