

# ***CS200 - Computer Organization***

## **Data Internals - Part2**

Aravind Mohan

Allegheny College

October 13, 2022



# Motivation to learn data representation

- How is data represented internally?
- How is data represented and processed internally, while executing an operator?

# Multiplication and Division



```
1  #include <stdio.h>
2  int main(){
3      int alpha = 10;
4      int beta = 3;
5      int gamma = 5;
6      alpha *= beta;
7      alpha /= gamma;
8      printf("%d\n", alpha);
9  }
```

**Q<sub>1</sub>:** What happens when lines 6, 7, and 8 are executed?

# Shift operator



- Shift Left: Makes a number grow bigger.
- Shift Right: Makes a number smaller.
- Test it out:  $10 \ll 2$
- Test it out:  $10 \gg 2$

# Shift operator (c'ntd)

- $\alpha \ll 0 \implies X \times 1$
- $\alpha \ll 1 \implies X \times 2$
- $\alpha \ll 2 \implies X \times 4$
- $\alpha \ll 3 \implies X \times 8$

$\alpha$  is the binary representation of decimal  $X$

# Shift operator (c'ntd)

- $\alpha \gg 0 \implies \frac{X}{1}$
- $\alpha \gg 1 \implies \frac{X}{2}$
- $\alpha \gg 2 \implies \frac{X}{4}$
- $\alpha \gg 3 \implies \frac{X}{8}$

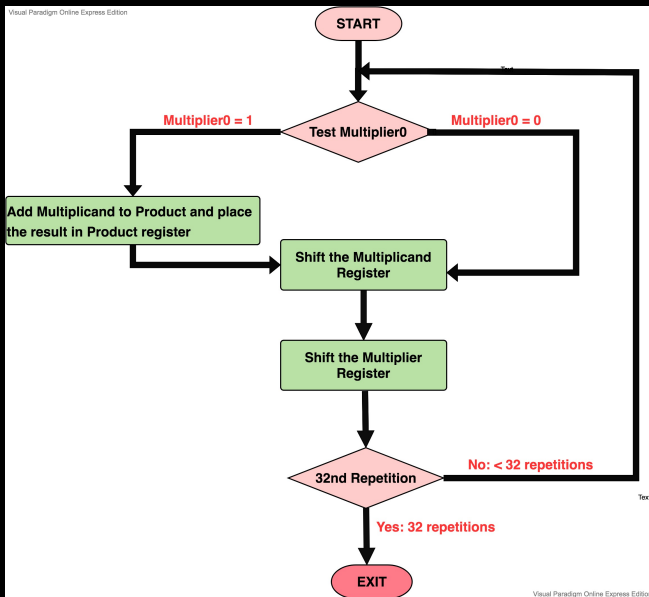
$\alpha$  is the binary representation of decimal X

# Multiplication - General Rules



- A process that executes repeated addition.
- Multiplying by any positive number, which is greater than 1, makes the number grow big. Similar to left shift operator.
- Test it out: can we multiply 15 by 2,3,4,5?

# Multiplication Algorithm





# Multiplication Example - 1

$$15 \times 7$$

Multiplicand	Multiplier	Product
0000 1111	0000 011 <b>1</b>	0000 0000
←	→	0000 1111
0001 1110	0000 001 <b>1</b>	
←	→	0010 1101
0011 1100	0000 000 <b>1</b>	
←	→	0110 1001
0111 1000	0000 000 <b>0</b>	
←	→	<b>0110 1001</b>

# Multiplication Example - 2

$$13 \times 9$$

Multiplicand	Multiplier	Product
0000 1101	0000 100 <b>1</b>	0000 0000
←	→	0000 1101
0001 1010	0000 010 <b>0</b>	
←	→	0000 1101
0011 0100	0000 001 <b>0</b>	
←	→	0000 1101
0110 1000	0000 000 <b>1</b>	
←	→	<b>0111 0101</b>

# Multiplication Example - 3

$$10 \times 13$$

Multiplicand	Multiplier	Product
0000 1010	0000 110 <sup>1</sup>	0000 0000
←	→	0000 1010
0001 0100	0000 011 <sup>0</sup>	
←	→	0000 1010
0010 1000	0000 001 <sup>1</sup>	
←	→	0011 0010
0101 0000	0000 000 <sup>1</sup>	
←	→	1000 0010

# Multiplication Try Out

- $5 \times 6$
- $7 \times 3$

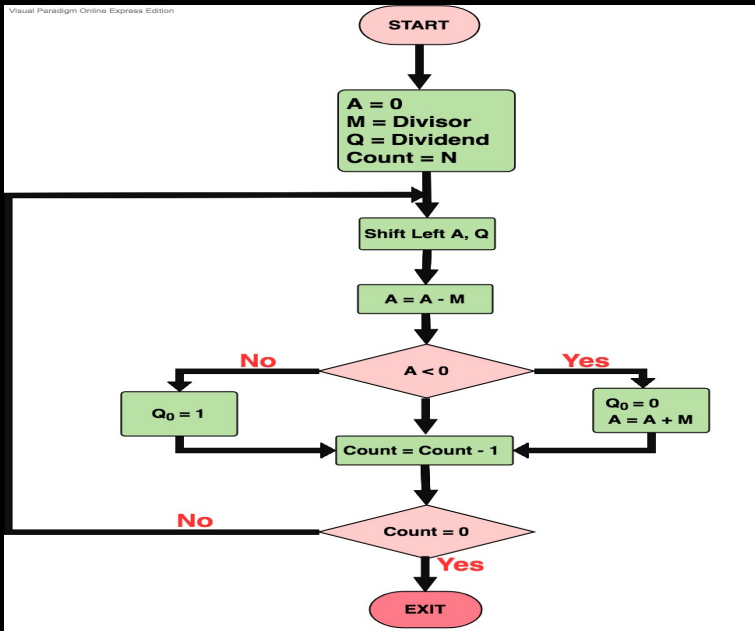
# Division - General Rules



- A process that executes repeated subtraction.
- Dividing by any positive number, which is greater than 1, makes the number smaller. Similar to right shift operator.
- Test it out: can we divide 32 by 2,4,8,16,32?

# Division Algorithm

Visual Paradigm Online Express Edition



# Division Example - 1

$$\frac{12}{5}$$

Operation	A	Q	M	Count
Initialize	00000	1100	00101	4
Left Shift	00001	100-	00101	4
Subtract	00001 + 11011 = 11100	1000	00101	4
Sum	11100 + 00101 = 00001	1000	00101	4
	00001	1000	00101	3

# Division Example - 1 (contd)

$$\frac{12}{5}$$

Left Shift	00011	000-	00101	3
Subtract	00011 + 11011 = 11110	0000	00101	3
Sum	00011	0000	00101	3
	<b>00011</b>	<b>0000</b>	<b>00101</b>	<b>2</b>
Left Shift	00110	000-	00101	2
Subtract	00110 + 11011 = 00001	0001	00101	2
	<b>00001</b>	<b>0001</b>	<b>00101</b>	<b>1</b>



# Division Example - 1 (contd)

$$\frac{12}{5}$$

Left Shift	00010	001-	00101	1
Subtract	00010 + 11011 = 11101	0010	00101	1
Sum	00010	0010	00101	1
	00010	0010	00101	0

# Let us try out some examples?

- $10/2$
- $13/5$
- $16/7$
- $17/6$

## Next ...

- Logic gates
- Assembly language programming

# Reading Assignment

- **PH** - chapter 03: [3.3];

# Questions

Do you have any questions from this class discussion?