**Q.1. Assume x=5, y=2, and z=3. Evaluate the value of each of the following Boolean expressions (show details).**

**a) [(x > 5) NAND (y < 5)] NOR ( y ≤ z)**

**Answer:-**

Let expression E = [(x > 5) NAND (y < 5)] NOR ( y ≤ z)

When x,y and z are substituted then

E = [(5 > 5) NAND (2 < 5)] NOR (2 <= 3)

E = [ false NAND true] NOR true

E = true NOR true

E = false.

**b) (x+y ≥ z ) XOR [( x>6) OR (z ≤ 5)]**

**Answer:-**

Let expression E = (x+y ≥ z ) XOR [( x>6) OR (z ≤ 5)]

When x,y and z are substituted then

E = (5 + 2 >= 3) XOR [(5 > 6 ) OR (3 <= 5)]

E = true XOR [false OR true]

E = true XOR true

E = false.

**c) [NOT( z >5) ] AND [(z=7) XOR (y<2)]**

**Answer:-**

Let expression E = [NOT( z >5) ] AND [(z=7) XOR (y<2)]

When x,y and z are substituted then

E = [NOT (3 > 5)] AND [(3 =7) XOR (2 < 2)]

E =[ NOT false ] AND [ false XOR false]

E = true AND false

E = false.

**Question 2**

**a)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **X** | **Y** | **Z** |
| **0** | **0** | **0** | **1** | **1** | **1** |
| **0** | **0** | **1** | **1** | **0** | **0** |
| **0** | **1** | **0** | **1** | **1** | **1** |
| **0** | **1** | **1** | **1** | **1** | **1** |
| **1** | **0** | **0** | **0** | **1** | **0** |
| **1** | **0** | **1** | **0** | **1** | **0** |
| **1** | **1** | **0** | **1** | **1** | **1** |
| **1** | **1** | **1** | **1** | **1** | **1** |

**Q.2.b)**

X = [ (NOT A) OR B ]

Y = [ A OR B OR ( NOT C ) ]

Z = [( NOT A ) OR B] AND [ A OR B OR ( NOT C )].

**Question 3**

**Truth table**

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **C** | **X** |
| **0** | **0** | **0** | **0** |
| **0** | **0** | **1** | **0** |
| **0** | **1** | **0** | **0** |
| **0** | **1** | **1** | **1** |
| **1** | **0** | **0** | **0** |
| **1** | **0** | **1** | **1** |
| **1** | **1** | **0** | **1** |
| **1** | **1** | **1** | **1** |

**Expression :-**

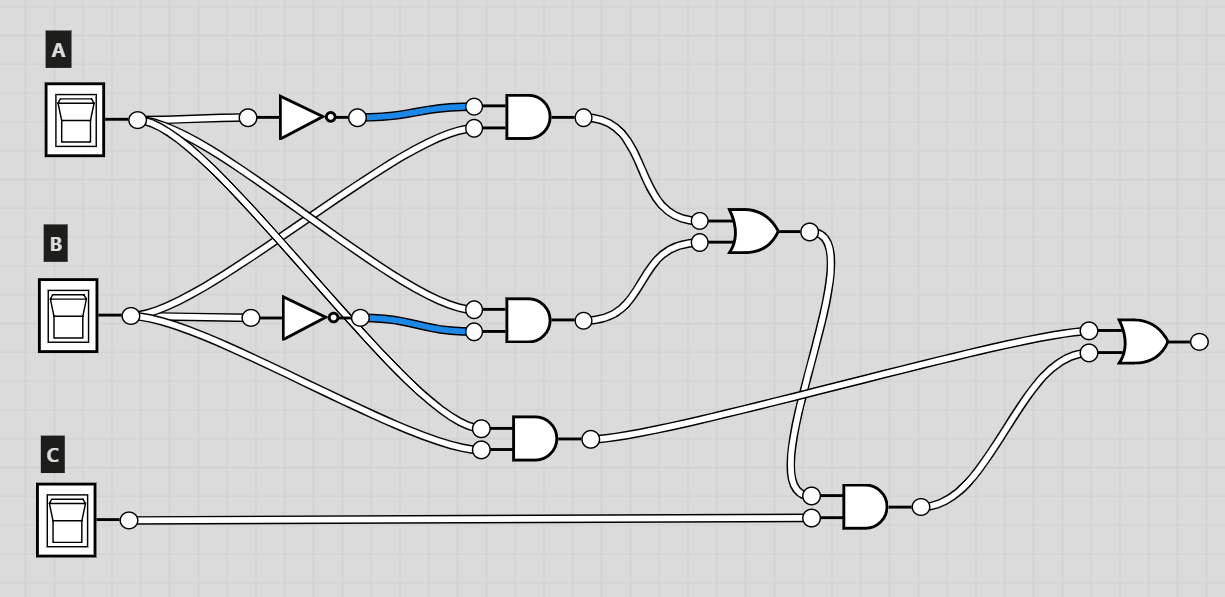
USING SUM OF PRODUCTS ALGORITHM THE EXPRESSION IS :

[(NOT A ). B. C] + [(A.NOT) B.C] + [A.B.(NOT C)] + (A.B.C)

C.[(NOT A) . B + A. (NOT B)] + A.B . ( NOT C + C)

C.[(NOT A) .B + A. (NOT B)] + A. B.

**Circuit**



**Question 4**

**Convert the natural integer 2018 in decimal to (show calculations):**

**a) Binary**

**111111000102**

**b) Base 4**

**1332024**

**c) Octal**

**37428**

**d) Hexadecimal**

**7E216**

**Question 5**

**Perform the following operations as follows**

**a) 78 – 120 in a sign and magnitude binary representation**

**Assuming 8 bit integers and adding gives**

7810 = 10011102

-12010 = 111110002

---------------------------------------------

(1)010001102

Which is equal to 7010

**a) 62 + 98 in unsigned 8-bit integer**

6210 = 1111102

9810 = 11000102

-------------------------

101000002 (sum which is 160 in binary)

**b) 34 - 78 in a signed two’s complement 8-bit representation**

Two's complement for a positive number is same as the binary number so

342  = 00100010

Two’s complement for a -ve number is 1’s complement + 1

-782 = 01001110 in binary

= 10110001 1’s complement

= 10110010 2’s complement by adding 1 to the above

Adding the 2 numbers

00100010

10110010

-----------------

**1**1010100

-----------------

**1**0101011 -- flip the bits other than the highest significant digit

1 -- add

----------------

**1**0101100

Which is equal to **-44**

**c) (F9A)16 + (7BB2)16 in hexadecimal**

F9A16

7BB216

------------

8B4C 16

------------

**d) (654)8 + (566)8 in octal**

6548

5668

-----------

14428

-----------

**e) (F3A)16 - (BFF)16 in hexadecimal**

F3A16

- BFF16

**---------------------**

33B

-------------

**f) (4111)5 - (3322)5 in base 5**

41115

- 33225

-------------

02345

--------------

**Question 6**

**What is the minimum number of bits needed to represent the following? (justify your answer)**

**a) (512)10**

51210 when represent in binary becomes

1000000000

Minimum bits required 10.

**b) (320)10**

32010 when represents in binary becomes

101000000

Minimum bits required 9.

**c) (3DE)16**

3DE10 when represents in binary becomes

1111011110

Minimum bits required 10.

**d) (67)8**

678 when represents in binary becomes

1000011

Minimum bits required 7.

**Question 7**

**a) 44 + 101**

4410 = 001011002

10110 = 011001012

------------------------

14510 = 100100012

------------------------

**b) 32 - 12**

3210 = 001000002

1210 = 000011002

--------------------------

1’s = 11110011

1

--------------------------

11110100 (-12 in 2’s complement notation)

Adding 32 + (-12)

00100000

11110100

---------------------------

1. 00010100 discard the carry over and the result 20 .

**c) -122 -15**

**-122** = 01111010

-------------------------

1’s 10000101

1

-------------------------

10000110 (-122 in 2’s complement notation)

--------------------------

**-15**  = 00001111

--------------------------

1’s 11110000

1

--------------------------

11110001 (-15 in 2’s complement notation)

----------------------------

10000110

11110001

----------------------------

1. 01110111 -- overflow and carry over

**Answer is :** 199which is wrong..

**Question 8**

**Consider the IEEE standard for floating-point numbers.**

**a) Explain how the number of bits used for the mantissa and exponent relates to the range the precision of floating-point numbers.**

As the number of bits to store exponent increase, then the range of the floating point number increases, as it as more bits to store the integer part. On the other hand as we have the fixed number of bits, the number of bits for mantissa decrease resulting in the losing of precision bit.

**b) Convert the decimal number -527.123 into the IEEE standard 32-bit format for floating-point numbers.**

A IEEE 32 bit format floating number is represented as flow

|  |  |  |
| --- | --- | --- |
| 1 bit (for sign) | 8 bits (for exponent) | 23 bits (mantissa) |

Sign Part:

Since it a negative number the sign bit is 1

Converting Integral part to binay:

52710 = 10 0000 11112

Converting the fraction part 0.123 in to binary

0.123 \* 2 = 0.246 + 0

0.246 \* 2 = 0.492 + 0

0.492 \* 2 = 0.984 + 0

0.984 \* 2 = 0.968 + 1

0.968 \* 2 = 0.936 + 1

0.936 \* 2 = 0.872 + 1

0.872 \* 2 = 0.744 + 1

0.744 \* 2 = 0.488 + 1

0.488 \* 2 = 0.976 + 0

0.976 \* 2 = 0.952 + 1

0.952 \* 2 = 0.904 + 1

0.904 \* 2 = 0.808 + 1

0.808 \* 2 = 0.616 + 1

0.616 \* 2 = 0.232 + 1

0.232 \* 2 = 0.464 + 0

0.464 \* 2 = 0.928 + 0

0.928 \* 2 = 0.856 + 1

Full number in binary number is:

527.12310 = 10 0000 1111.0001 1111 0111 1100 1

Normalize the binary representation of the number:

527.12310 = 10 0000 1111.0001 1111 0111 1100 12

= 10 0000 1111.0001 1111 0111 1100 12 \* 20

= 1.0 0000 1111 0001 1111 0111 1100 12 \* 29

Add bias to the exponent of 2

Effective Exponent = 9 + bias

= 9 + 28-1 - 1

= 13610

= 1000 10002

Normalize Mantissa (remove left leading one and adjust its length to 23 bits)

Mantissa = 1.0 0000 1111 0001 1111 0111 1100 1

Normalised Mantissa = 000 0011 1100 0111 1101 1111

Final Floating point representation:

|  |  |  |
| --- | --- | --- |
| 1 | 1000 1000 | 000 0011 1100 0111 1101 1111 |

**Question 9**

**Assume you have a 100x100 pixels RGB coloured image where each pixel has three colour components (red, green, and blue), and the range of each colour is [0 255]:**

**a) How is this image represented internally as zeroes and ones?**

The colour of each pixel is mapped to three component colours using the binary representation (0s and 1s) of the each colour ranging from 0-255.

Ex: yellow is (255, 255,0) and the binary representation of the that can be

|  |  |  |
| --- | --- | --- |
| 1111 1111 | 1111 1111 | 0000 0000 |

In a similar way any other pixel colour can be represented in this pattern

**b) How many bytes does it take to store it?**

Number of bits required to store a particular colour in red, green or blue = 8 bits

Number of bits to store a colour of pixel = 8 \* 3 = 24 bites

Number of bits required to store the image = 24 \* 100 \* 100 = 240000 bits

Number of bytes required = 240000 / 8 = 30000 bytes

**Question 10:**

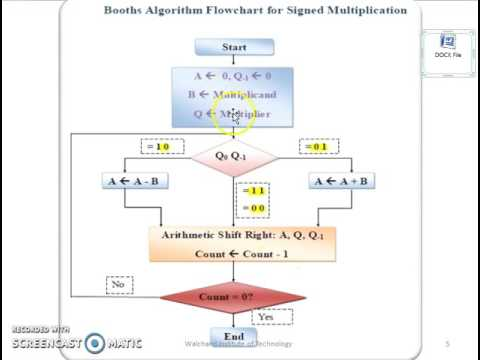
**a) Booth's multiplication algorithm is an algorithm to multiply two numbers in two's complement notation. Explain the algorithm and demonstrate how it works using one example.**

**Example:**

**Let Q = 01002 be multiplier**

**Let B = 01112 be multiplicand**

**-B in 2’s complement = 10012**

****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Iteration** | **Step** | **A** | | **Qn-1** | **Description** |
| **0** | **0** | **0 0 0 0** | **0 1 0 0** | **0** | **Initialization** |
| **1** | **1.00** |  |  |  | **No Action** |
| **1** | **2** | **0 0 0 0** | **0 0 1 0** | **0** | **Shift** |
| **2** | **1.00** |  |  |  | **No Action** |
| **2** | **2** | **0 0 0 0** | **0 0 0 1** | **0** | **Shift** |
| **3** | **1.10** | **1 0 0 1** | **0 0 0 1** | **0** | **Subtraction** |
| **3** | **2** | **1 1 0 0** | **1 0 0 0** | **1** | **Shift** |
| **4** | **1.01** | **0 0 1 1** | **1 0 0 0** | **1** | **Add** |
| **4** | **2** | **0 0 0 1** | **1 1 0 0** | **0** | **Shift** |

The production is 0001 11002  which is 28

**Reference:** [**https://www.geeksforgeeks.org/computer-organization-booths-algorithm/**](https://www.geeksforgeeks.org/computer-organization-booths-algorithm/)

**b) In the context of risk & safety when opening a computer, explain the risk of static electricity. What should you do to avoid any damage to yourself or the components of a computer?**

The internal parts of the computer and essentially the hard drive are extremely susceptible to static electricity, which can cause considerable damage to the hard drive, if it is zapped with even a small amount.

Wearing a electrostatic discharge wrist strap or working on an anti-static matt will prevent any static electricity from damaging your computer.

**References :-**

<https://www.webopedia.com/DidYouKnow/Computer_Science/static.asp>

**c) In the context of representing floating-point numbers in binary format, explain what is meant by the following (give examples of each case). - Range - Precision - Accuracy**

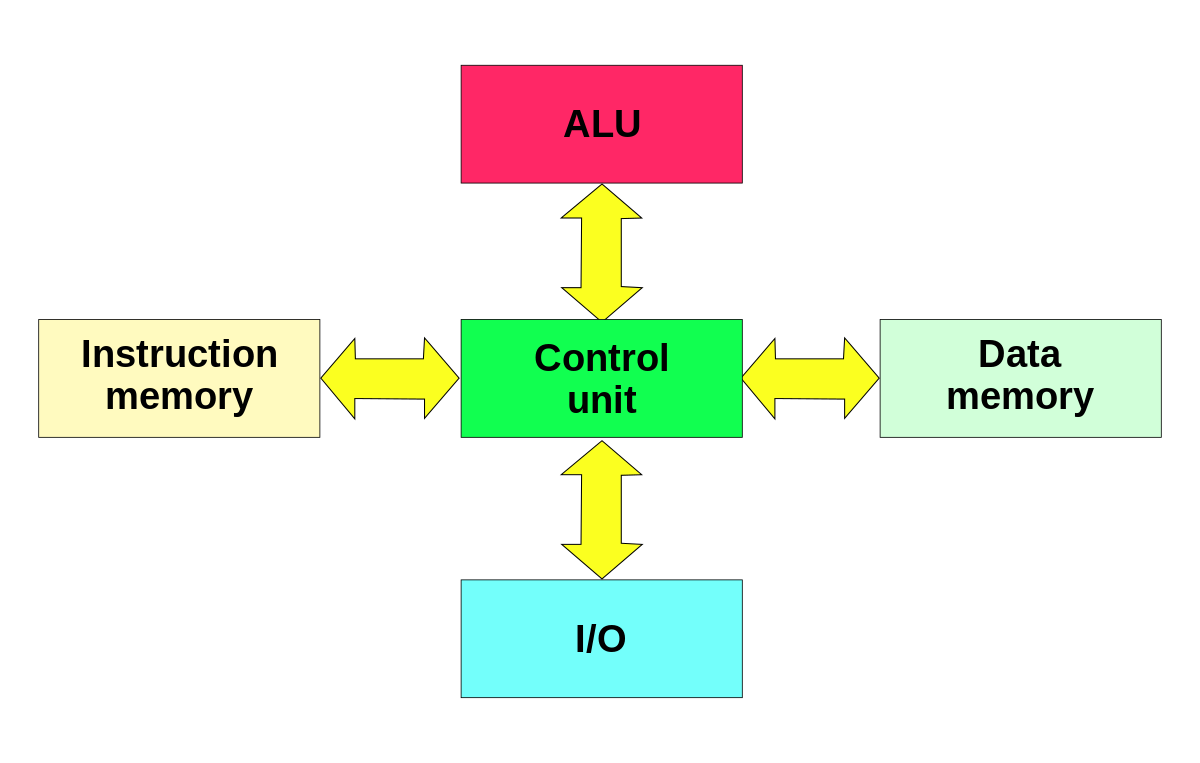
Range: In a floating point representation range defines the min and max values possible for a particular representation. The range is limited by the exponent.

Precision: The smallest change that can be represented in floating point representation is called as precision.

Accuracy: Accuracy in floating point representation is governed by number of significant bits, whereas range is limited by exponent

<https://www.geeksforgeeks.org/floating-point-representation-basics/>

**d) In the lectures, we have been focussing on the Von Neumann architecture. However, this is not the only architecture. Explain the Harvard architecture and highlight the main differences with the Von Neumann one.**



The Harvard architecture stores machine instructions and data in separate memory units that are connected by different busses.

In this case, there are at least two memory address spaces to work with, so there is a memory register for machine instructions and another memory register for data.

Computers designed with the Harvard architecture are able to run a program and access data independently, and therefore simultaneously.

Harvard architecture has a strict separation between data and code.

Harvard architecture is more complicated but separate pipelines remove the bottleneck that Von

Neumann creates

|  |  |
| --- | --- |
| [**Harvard architecture**](http://www.polytechnichub.com/difference-harvard-architecture-von-neumann-architecture/) | [**Von Neumann architecture**](http://www.polytechnichub.com/difference-harvard-architecture-von-neumann-architecture/) |
| Harvard architecture – diagram  Harvard architecture | Von Neumann architecture – diagram  Von Neumann architecture |
| The name is originated from “Harvard Mark I” a relay based old computer. | It is named after the mathematician and early computer scientist John Von Neumann. |
| It required two memories for their instruction and data. | It required only one memory for their instruction and data. |
| Design of Harvard architecture is complicated. | Design of the von Neumann architecture is simple. |
| Harvard architecture is required separate bus for instruction and data. | Von Neumann architecture is required only one bus for instruction and data. |
| Processor can complete an instruction in one cycle | Processor needs two clock cycles to complete an instruction. |
| Easier to pipeline, so high performance can be achieve. | Low performance as compared to Harvard architecture. |
| Comparatively high cost. | It is cheaper. |

Reference: <http://www.differencebetween.net/technology/difference-between-von-neumann-and-harvard-architecture/>

<https://www.microcontrollertips.com/whats-the-difference-between-von-neumann-and-harvard-architectures/>