CS101 Introduction to Computing
Lec-3
Numbers, Combinational Logic Circuits & Beyond

Friday, March 12, 2021

How to Encode Numbers: Binary Numbers

- Working with binary numbers
  - In base ten, helps to know powers of 10
    - One, Ten, Hundred, Thousand, ...
  - In base two, helps to know powers of 2
    - One, Two, Four, Eight, Sixteen, ..
    - Count up by powers of two

2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
512	256	128	64	32	16	8	4	2	1

Important Property of Binary Number

- Number of different numbers can be possible for a N-bit binary number
  - 2^N, for 2 bit number it is 4 (00, 01, 10 and 11)

Octal (Base 8)

- Shorter & easier to read than binary
- 8 digits: 0, 1, 2, 3, 4, 5, 6, 7
- **Octal numbers to Decimal**

$$
\begin{aligned}
136_8 &= 1*8^2 + 3*8^1 + 6*8^0 \\
&= 1*64 + 3*8 + 6*1 \\
&= 64 + 24 + 6 \\
&= 94_{10}
\end{aligned}
$$

Hexadecimal (base 16)

- Shorter & easier to read than binary
- 16 digits:

  0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
- "0x" often precedes hexadecimal numbers

$$
\begin{aligned}
0x123 &= 1*16^2 + 2*16^1 + 3*16^0 \\
&= 1*256 + 2*16 + 3*1 \\
&= 256 + 32 + 3 \\
&= (291)_{10}
\end{aligned}
$$

Counting

Dec	Binary	Oct	Hex	Dec	Binary	Oct	Hex
0	00000	0	0	8	01000	10	8
1	00001	1	1	9	01001	11	9
2	00010	2	2	10	01010	12	A
3	00011	3	3	11	01011	13	B
4	00100	4	4	12	01100	14	C
5	00101	5	5	13	01101	15	D
6	00110	6	6	14	01110	16	E
7	00111	7	7	15	01111	17	F
8	01000	10	8	16	10000	20	10

Fractional Number

- Point:

  Decimal Point, Binary Point, Hexadecimal point

- Decimal

  $247.75 = 2 \times 10^{2} + 4 \times 10^{1} + 7 \times 10^{0} + 7 \times 10^{-1} + 5 \times 10^{-2}$

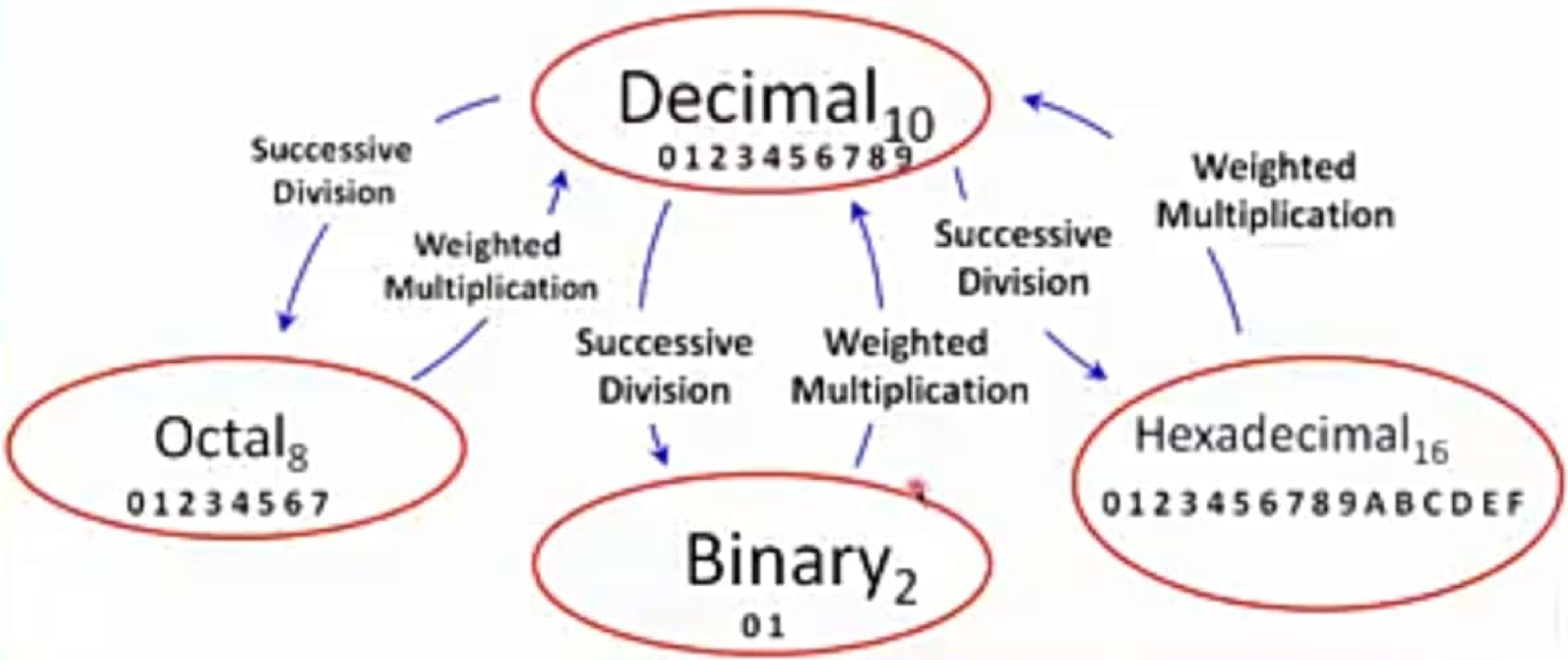
- Binary

  $10.101 = 1 \times 2^{1} + 0 \times 2^{0} + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$

- Hexadecimal

  $6A.7D = 6 \times 16^{1} + 10 \times 16^{0} + 7 \times 16^{-1} + D \times 16^{-2}$

Converting To and From Decimal



Decimal ↔ Binary

$Base_{10}$ DECIMAL → **Successive Division** → **$Base_2$** BINARY

a) Divide the decimal number by 2; the remainder is the LSB of the binary number.

b) If the quotient is zero, the conversion is complete. Otherwise repeat step (a) using the quotient as the decimal number. The new remainder is the next most significant bit of the binary number.

$Base_2$ BINARY → **Weighted Multiplication** → **$Base_{10}$** DECIMAL

a) Multiply each bit of the binary number by its corresponding bit-weighting factor (i.e., Bit-0→2^0=1; Bit-1→2^1=2; Bit-2→2^2=4; etc).

b) Sum up all of the products in step (a) to get the decimal number.

Decimal to Binary : Division Method

- Divide decimal number by 2 and insert remainder into new binary number.
  - Continue dividing quotient by 2 until the quotient is 0.
- Example: Convert decimal number 12 to binary

12 div 2 = (Quo=6 , Rem=0) LSB

6 div 2 = (Quo=3, Rem=0)

3 div 2 = (Quo=1,Rem=1)

1 div 2 = (Quo=0, Rem=1) MSB

$$\boxed{12_{10} = 1\,1\,00_{2}}$$

Decimal to Octal Conversion

The Process: Successive Division

- Divide number by **8**; R is the LSB of the **octal** number
- While Q is **not zero**
  - Using the Q as the decimal number. Divide
  - New remainder is MSB of the **octal** number.

$8\,\overline{)\,94}$ → quotient 11 $r = 6 \leftarrow$ LSB

$8\,\overline{)\,11}$ → quotient 1 $r = 3$

$8\,\overline{)\,1}$ → quotient 0 $r = 1 \leftarrow$ MSB

$$\boxed{94_{10} = 136_8}$$

Decimal to Hexadecimal Conversion

The Process: Successive Division

- Divide number by **16**; R is the LSB of the **hex** number
- While Q is NOT zero
  - Use the Q as the decimal number. Divide by 16
  - New remainder is MSB of the **hex** number.

$$\begin{array}{r} 5 \\ 16\overline{)\,94} \end{array} \quad r = E \leftarrow \text{LSB}$$

$$\begin{array}{r} 0 \\ 16\overline{)\,5} \end{array} \quad r = 5 \leftarrow \text{MSB}$$

$$\boxed{94_{10} = 5E_{16}}$$

Substitution Code

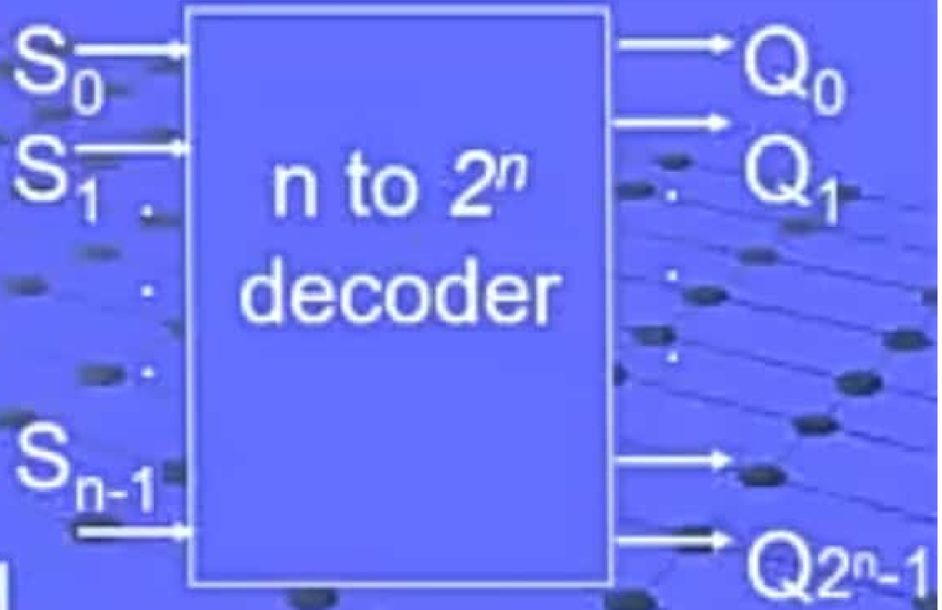
Convert $1110\ 0110\ 1010_2$ to hex using the 4-bit substitution code :

$$\overset{E}{\overbrace{1110}}\ \overset{6}{\overbrace{0110}}\ \overset{A}{\overbrace{1010}}$$

$$\boxed{E6A_{16}}$$

Decoder

A combinational circuit which has n inputs and 2^n outputs

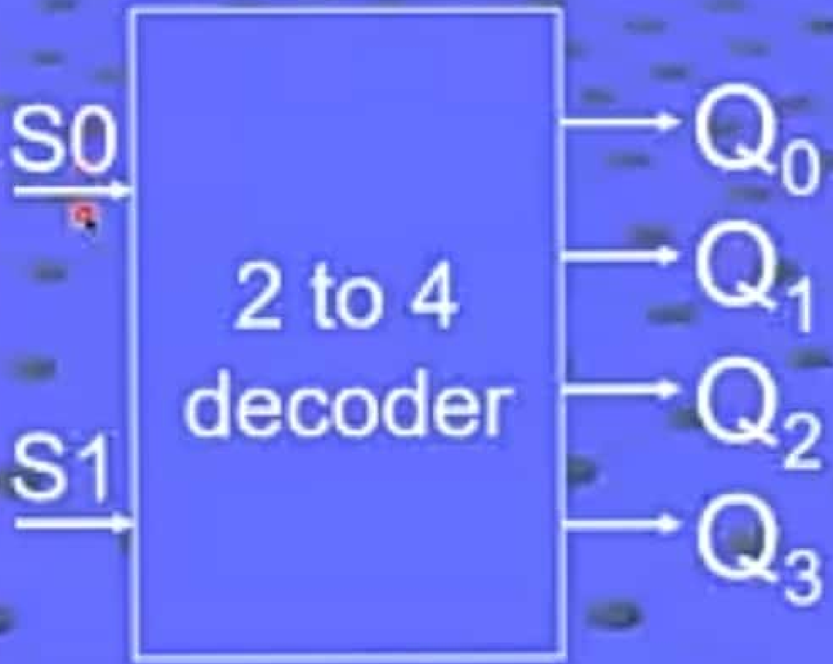


For an input combination the corresponding output becomes activate while all others remain inactive

It is used to select one out of 2^n entities

Friday, March 12, 2021

A 2 to 4 line Decoder

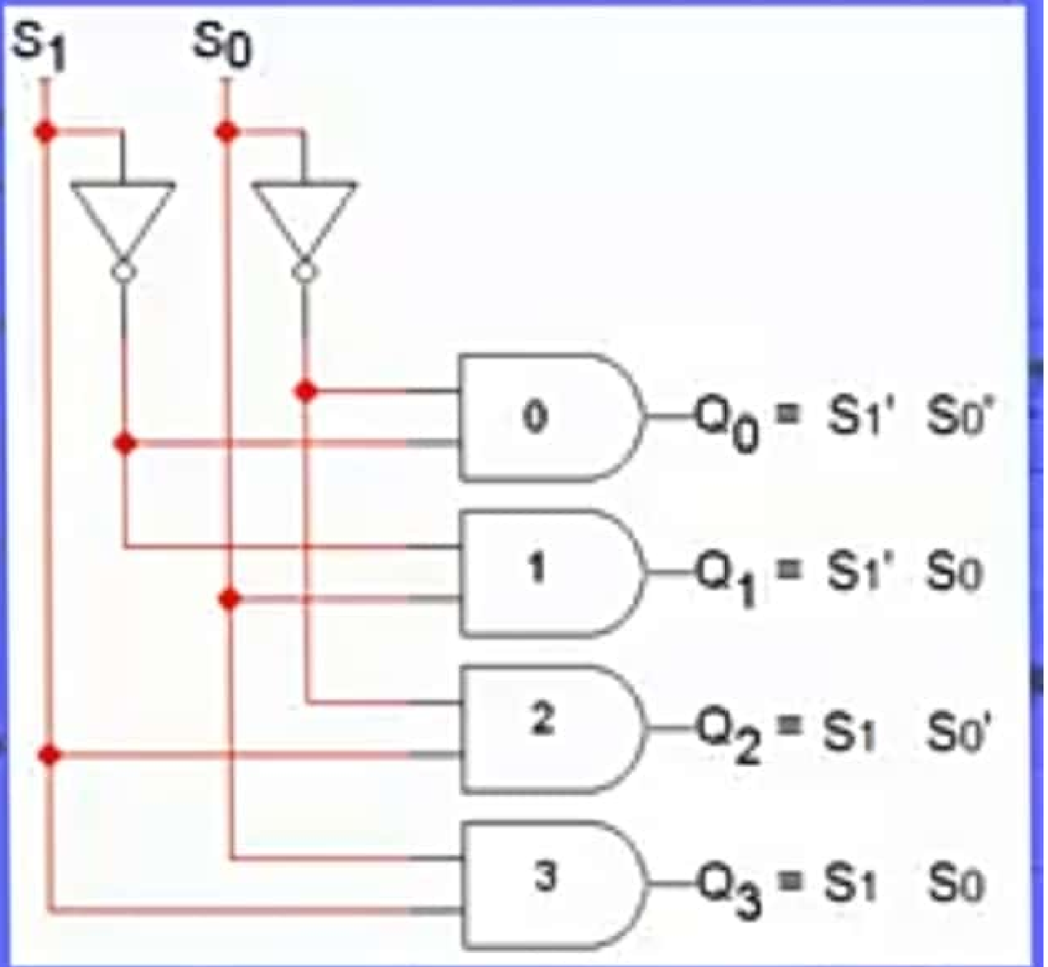


S1	S0		Q_3	Q_2	Q_1	Q_0
0	0		0	0	0	1
0	1		0	0	1	0
1	0		0	1	0	0
1	1		1	0	0	0

2-to-4 decoder

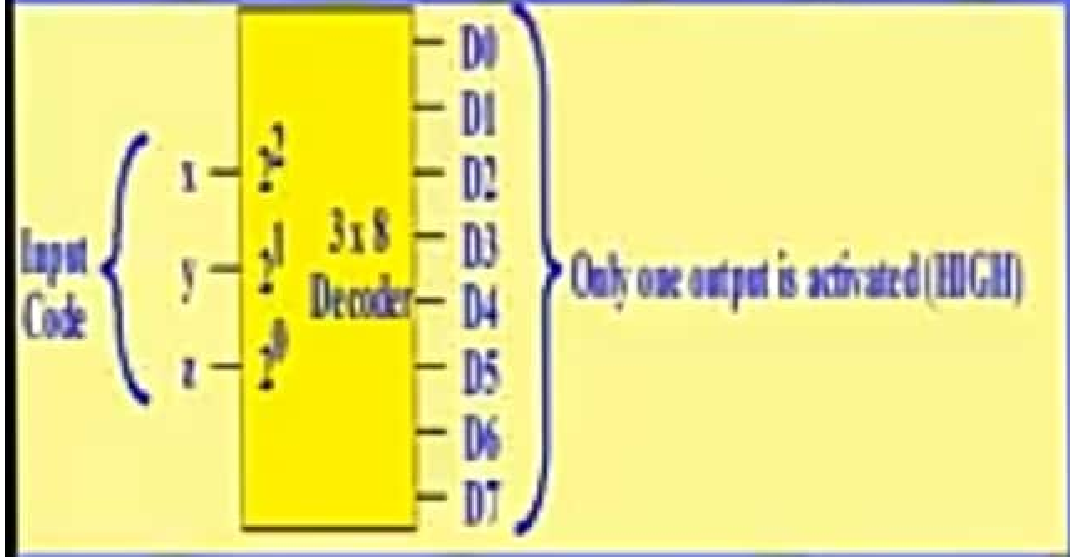
2 inputs and 2^2 *outputs*

S1	S0		Q_3	Q_2	Q_1	Q_0
0	0		0	0	0	1
0	1		0	0	1	0
1	0		0	1	0	0
1	1		1	0	0	0

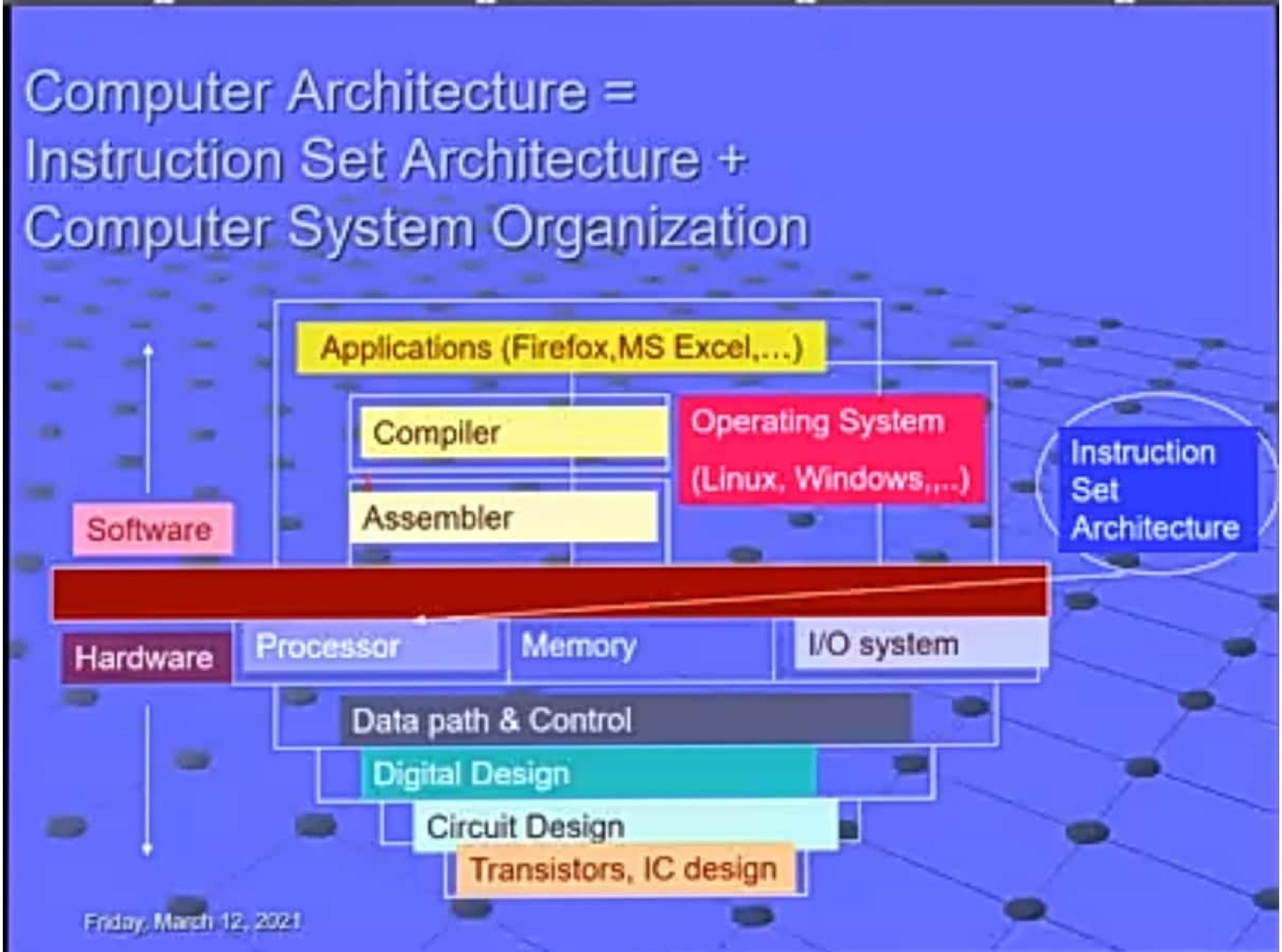


Observe that the output Q which becomes active has the subscript equal to decimal value of the input $(S1,S0)_{10}$

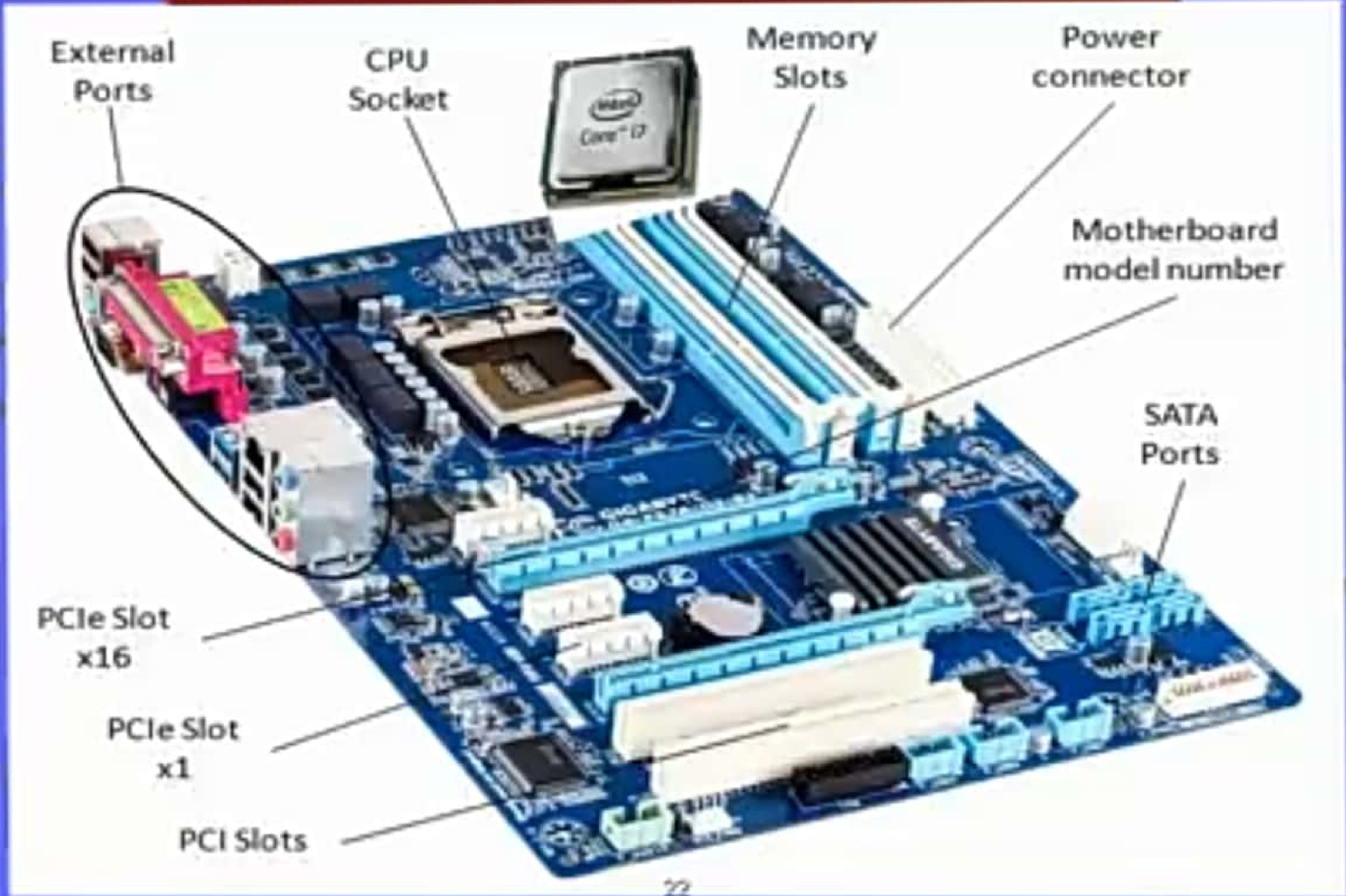
A 3x8 Decoder



x	y	z	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1


Computer Architecture =
Instruction Set Architecture +
Computer System Organization
Applications (Firefox,MS Excel,...)
Compiler
Assembler
Operating System
(Linux, Windows,,..)
Instruction
Set
Architecture
Software
Hardware
Processor
Memory
I/O system
Data path & Control
Digital Design
Circuit Design
Transistors, IC design
Friday, March 12, 2021

Inside PC : Motherboard



Inside PC : Processor

Inside PC: Memory Card