IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

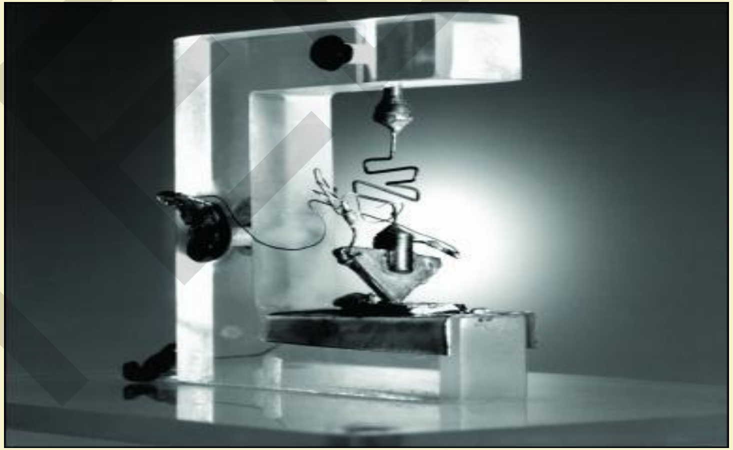
Digital Circuits

Introduction

Santanu Chattopadhyay
Electronics and Electrical Communication Engineering

The Start of the Modern Electronics Era

Bardeen, Shockley, and Brattain at Bell Labs - Brattain and Bardeen invented the bipolar transistor in 1947.



The first germanium bipolar transistor. Roughly 50 years later, electronics account for 10% (4 trillion dollars) of the world GDP.

Electronics Milestones

1874 Braun invents the solid-state rectifier.

1906 DeForest invents triode vacuum tube.

1907-1927 First radio circuits developed from diodes and triodes.

1925 Lilienfeld field-effect device patent filed.

1947 Bardeen and Brattain at Bell Laboratories invent bipolar transistors.

1952 Commercial bipolar transistor production at Texas Instruments.

1956 Bardeen, Brattain, and Shockley receive Nobel prize.

1958 Integrated circuit developed by Kilby and Noyce

1961 First commercial IC from Fairchild Semiconductor

1963 IEEE formed from merger or IRE and AIEE

1968 First commercial IC opamp

1970 One transistor DRAM cell invented by Dennard at IBM.

1971 4004 Intel microprocessor introduced.

1978 First commercial 1-kilobit memory.

1974 8080 microprocessor introduced.

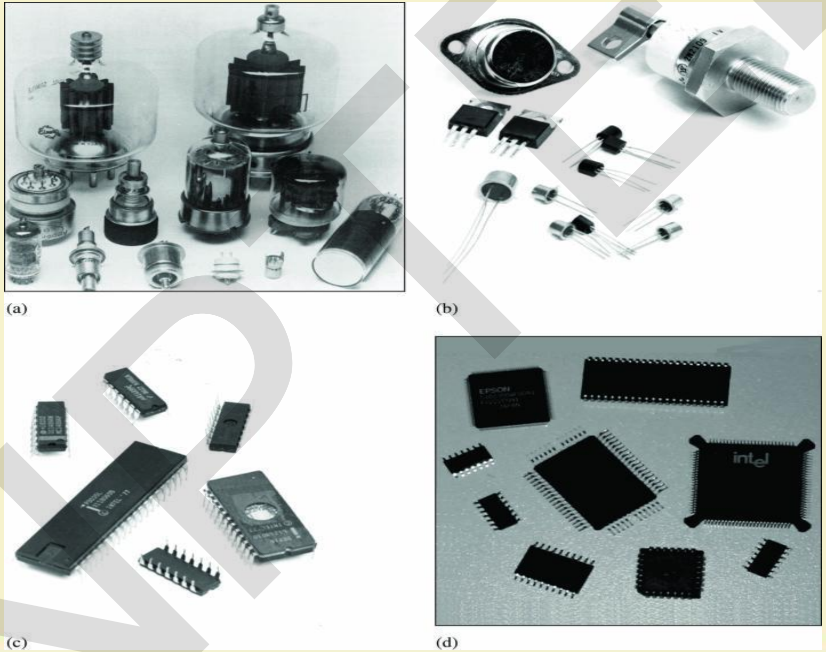
1984 Megabit memory chip introduced.

2000 Alferov, Kilby, and Kromer share Nobel prize

Evolution of Electronic Devices

Vacuum Tubes

Discrete Transistors



SSI and MSI Integrated Circuits

VLSI Surface-Mount Circuits

Microelectronics Proliferation

- The integrated circuit was invented in 1958.
- World transistor production has more than doubled every year for the past twenty years.
- Every year, more transistors are produced than in all previous years combined.
- Approximately 10^9 transistors were produced in a recent year.
- Roughly 50 transistors for every ant in the world .

*Source: Gordon Moore's Plenary address at the 2003 International Solid State Circuits Conference.

5 Commendments

- Moore's Law : The number of transistors on a chip doubles annually
- Rock's Law : The cost of semiconductor tools doubles every four years
- Machrone's Law: The PC you want to buy will always be $5000
- Metcalfe's Law : A network's value grows proportionately to the number of its users squared

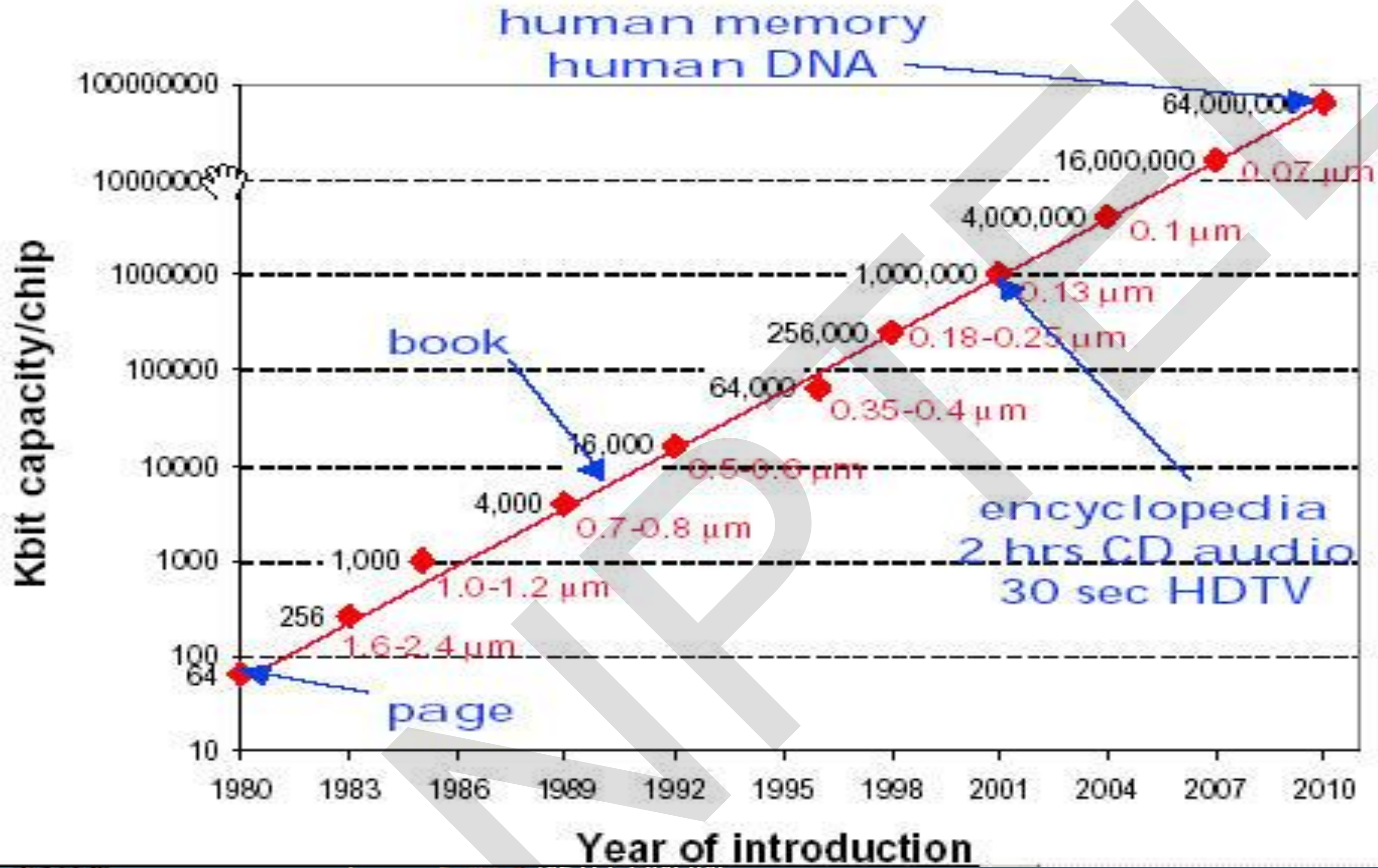
5 Commandments(cont.)

- Wirth’s Law : Software is slowing faster than hardware is accelerating
- Further Reading: “5 Commandments”, IEEE Spectrum December 2003, pp. 31-35.

Moore’s law

- Moore predicted that the number of transistors that can be integrated on a die would grow exponentially with time.
- Amazingly visionary – million transistor/chip barrier was crossed in the 1980’s.
- 16 M transistors (Ultra Sparc III)
- 140 M transistor (HP PA-8500)
- 1.7B transistor (Intel Montecito)

Evolution in DRAM Chip Capacity

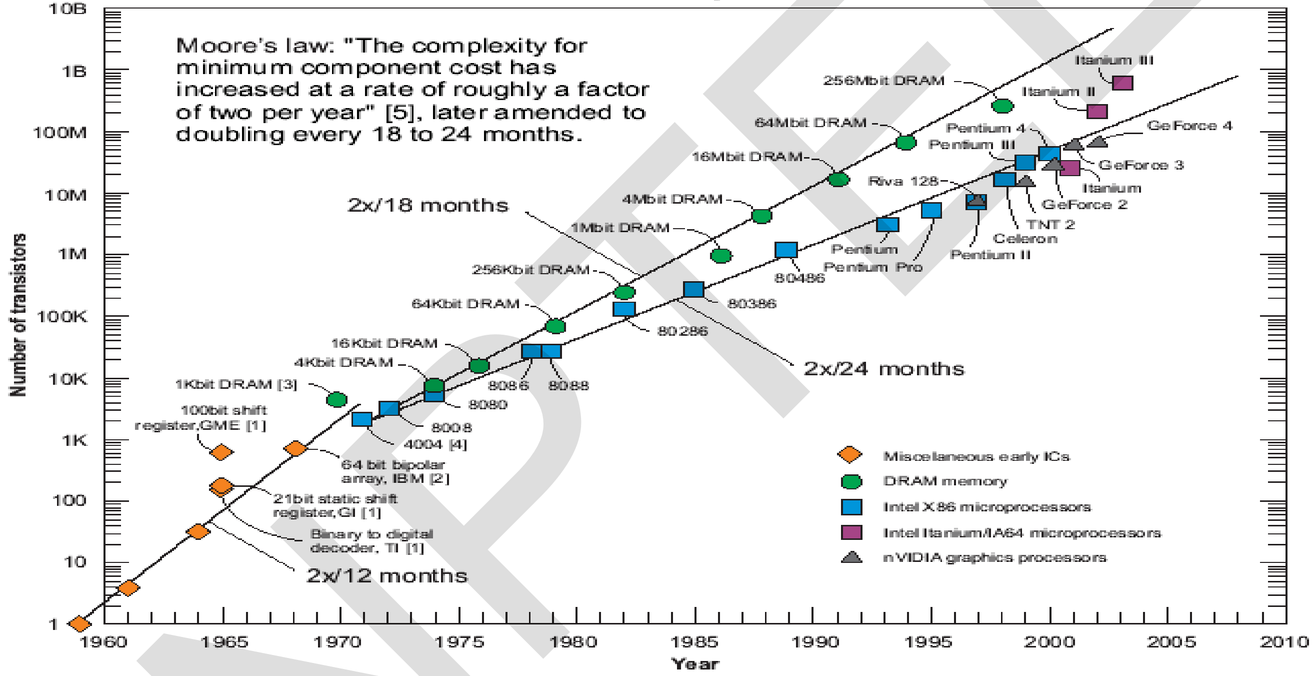


Memory IC (bits/chip)

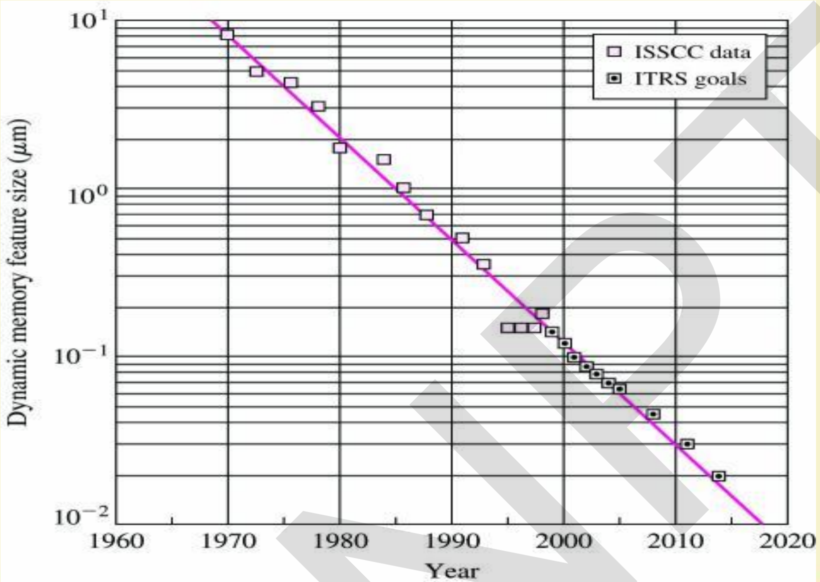
4 times every 3 years

Transistor Per Integrated Circuit Trends

www.icknowledge.com

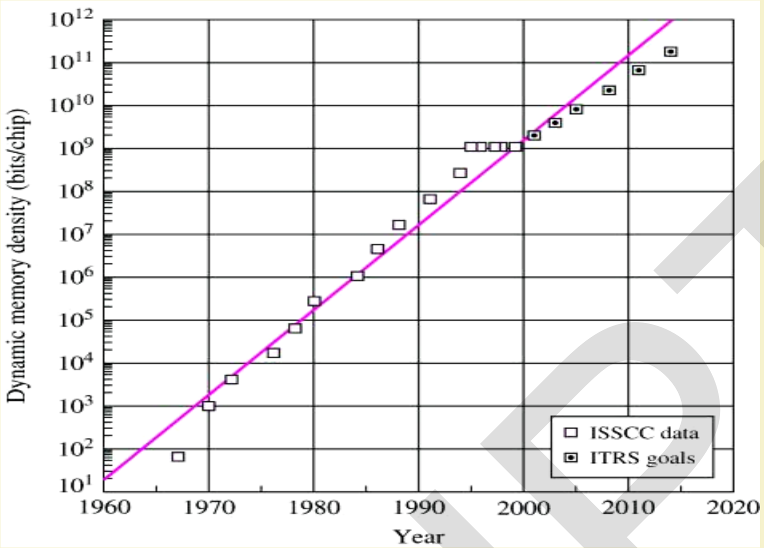


Device Feature Size

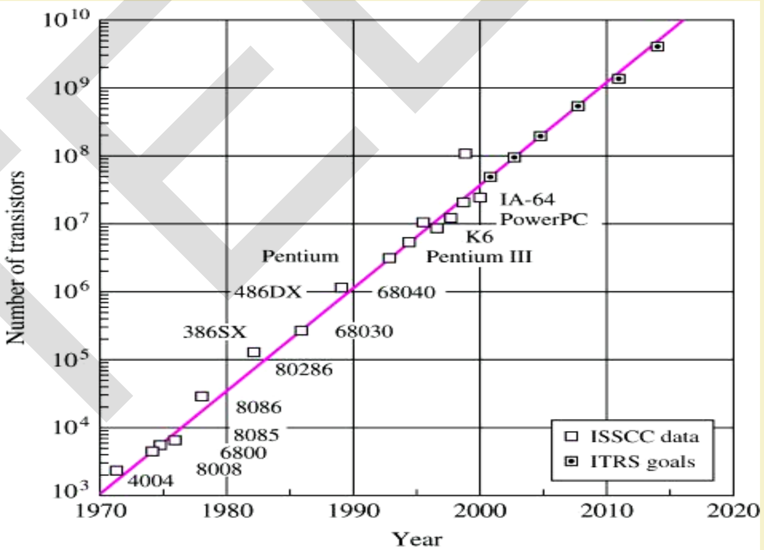


- Feature size reductions enabled by process innovations.
- Smaller features lead to more transistors per unit area and therefore higher density.

Rapid Increase in Density of Microelectronics



Memory chip density versus time.

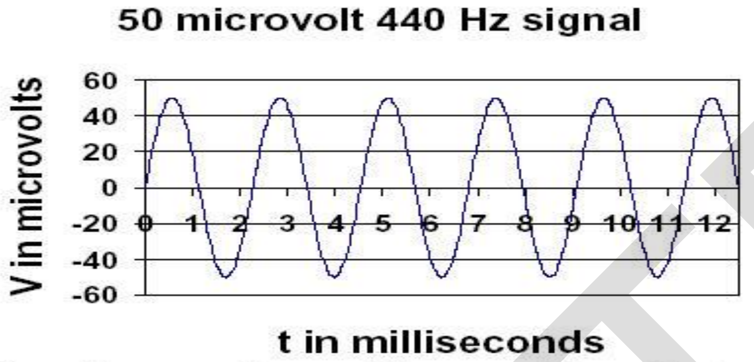


Microprocessor complexity versus time.

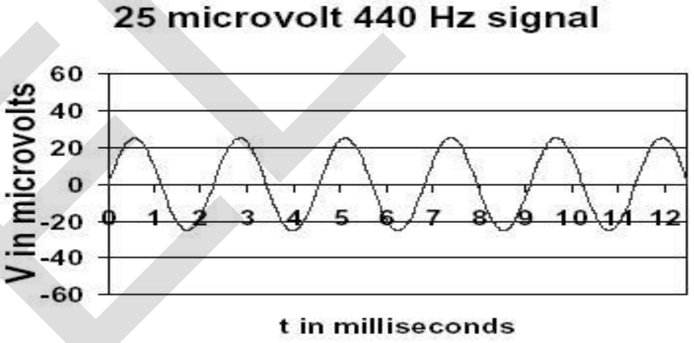
Analog versus Digital Electronics

- Most observables are analog
- But the most convenient way to represent and transmit information electronically is digital
- Analog/digital and digital/analog conversion is essential

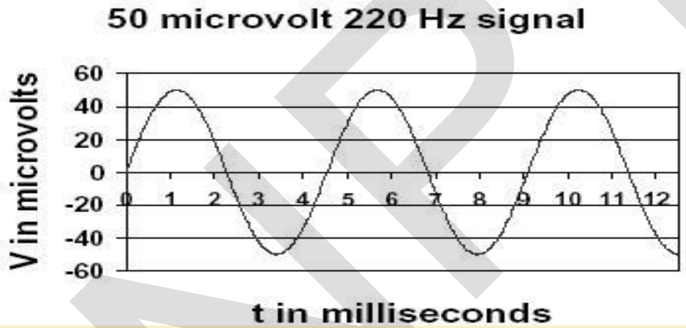
Analog signal (microphone voltage) representing piano key A (440 Hz)..



Microphone voltage with normal key stroke



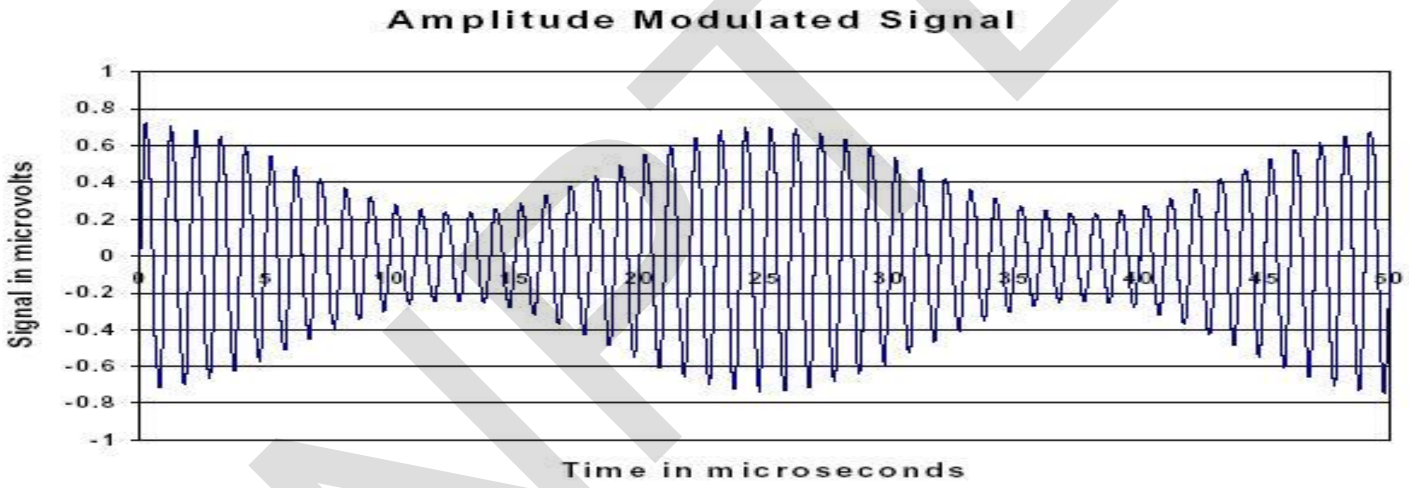
Microphone voltage with soft pedal

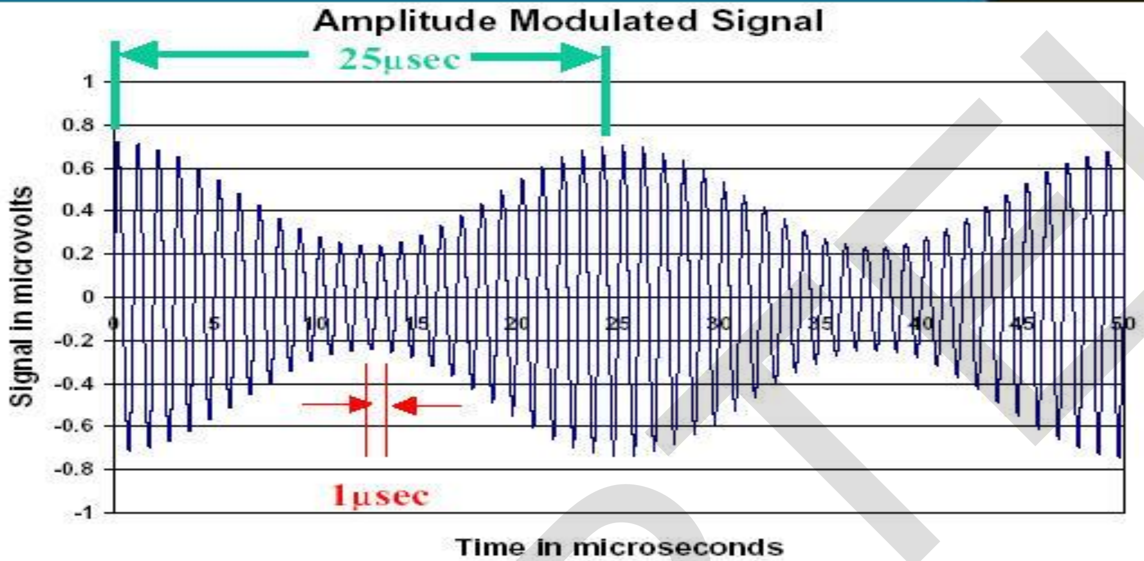


Analog signal representing piano A, but one octave below middle A (220 Hz)

Analog Signals

- May have very physical relationship to information presented
- In the simplest, are direct waveforms of information vs time
- In more complex cases, may have information modulated on a carrier as in AM or FM radio





Parameters of sin waves:

Period (time to repeat)

Frequency (1/period)

Phase

Amplitude

Note: The period of the carrier is 1µsec * (that is, the frequency is 1MHz)
The period of the modulation is 25µsec (that is, the frequency is 40kHz)
The amplitude of the modulation is about 50% of the maximum possible

What is the equation of this waveform (just for fun)??

Answer: V (in microvolts) = $[.5 + .25 \cos(2\pi \times 40 \times 10^3 t)] \times \sin(2\pi \times 10^6 t)$

Digital signal representation

- By using binary numbers we can represent any quantity.
- For example a binary two (10) could represent a 2 volt signal.
- We generally have to agree on some sort of "code" and the dynamic range of the signal in order to know the form and the minimum number of bits.
- Possible digital representation for a pure sine wave of known frequency.
  - We must choose maximum value and "resolution" or "error," then we can encode the numbers.
  - Suppose we want 1V accuracy of amplitude with maximum amplitude of 50V, we could use a simple pure binary code with 6 bits of information.

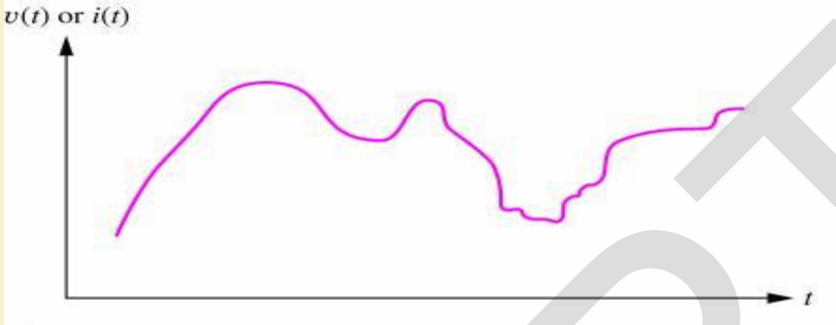
Digital representations of logical functions

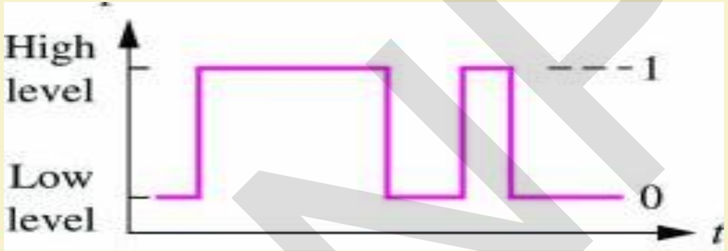
- Digital signals also offer an effective way to execute logic. The formalism for performing logic with binary variables is called switching algebra or Boolean algebra.
- Digital electronics combines two important properties:
  - The ability to represent real functions by coding the information in digital form.
  - The ability to control a system by a process of manipulation and evaluation of digital variables using switching algebra.

Digital Representations of logic functions (cont.)

- Digital signals can be transmitted, received, amplified, and retransmitted with no degradation.
- Binary numbers are a natural method of expressing logic variables.
- Complex logic functions are easily expressed as binary function.
- With digital representation, we can achieve arbitrary levels of " dynamic range," that is, the ratio of the largest possible signal to the smallest than can be distinguished above the background noise.
- Digital information is easily and inexpensively stored

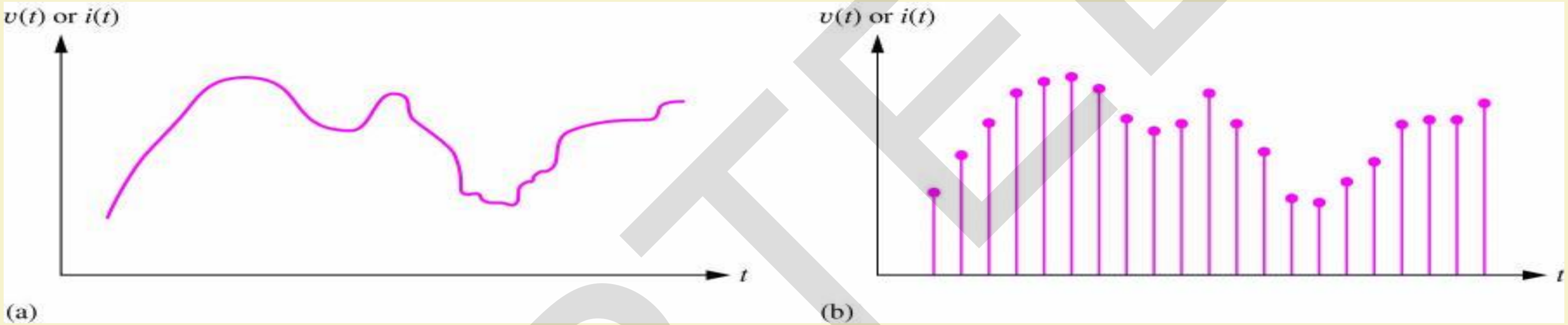
Signal Types





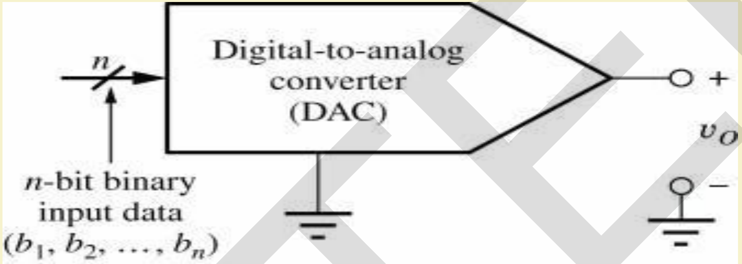
- Analog signals take on continuous values - typically current or voltage.
- Digital signals appear at discrete levels. Usually we use binary signals which utilize only two levels.
- One level is referred to as logical 1 and logical 0 is assigned to the other level.

Analog and Digital Signals



- Analog signals are continuous in time and voltage or current. (Charge can also be used as a signal conveyor.)
- After digitization, the continuous analog signal becomes a set of discrete values, typically separated by fixed time intervals.

Digital-to-Analog (D/A) Conversion



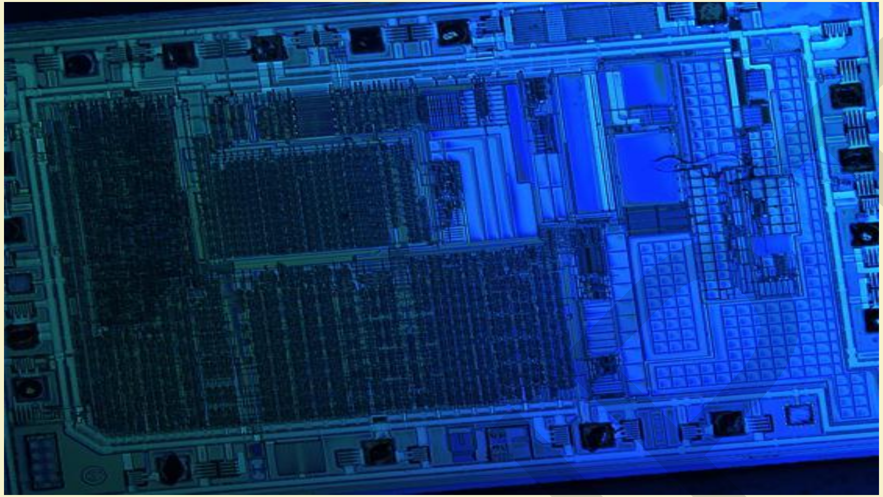
- For an n-bit D/A converter, the output voltage is expressed as:

$$V_O = (b_1 2^{-1} + b_2 2^{-2} + \ldots + b_n 2^{-n})V_{FS}$$

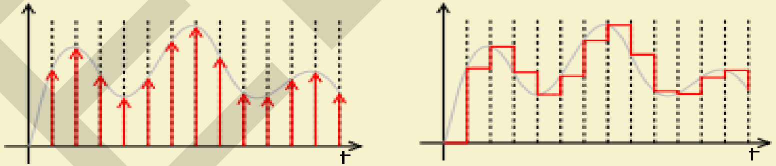
- The smallest possible voltage change is known as the least significant bit or LSB.

$$V_{LSB} = 2^{-n} V_{FS}$$

DAC

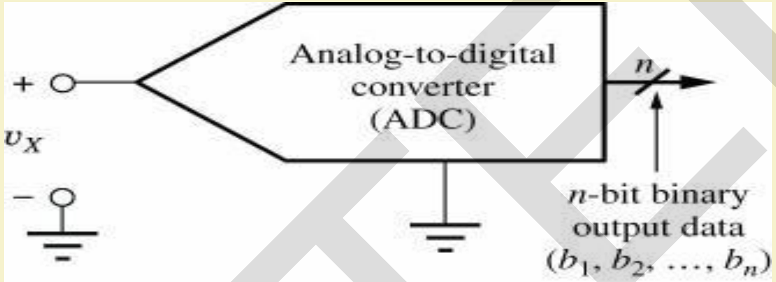


TI's 20-bit sigma delta DAC



8-channel digital-to-analog converter Cirrus Logic CS4382 placed on Sound Blaster X-Fi Fatal1ty

Analog-to-Digital (A/D) Conversion



- Analog input voltage v_x is converted to the nearest n-bit number.
- For a four bit converter, 0 -> v_x input yields a 0000 -> 1111 digital output.
- Output is approximation of input due to the limited resolution of the n-bit output. Error is expressed as:

$$V_{\varepsilon} = \left| v_x - (b_1 2^{-1} + b_2 2^{-2} + \ldots + b_n 2^{-n}) V_{FS} \right|$$

Analog and Digital Signals

We seem to live in an analogue world –

things can be louder or quieter, hotter or colder, longer or shorter, on a "sliding scale".

If we record sound on a tape recorder, we're putting an analogue signal onto the tape.

Digital signals aren't on a sliding scale – they're either ON or OFF. (We call these "1" and "0".) There's no "in between".

Analog and Digital Signals

Are these **analogue** or **digital**?

Volume control on a radio
Traffic lights
Motor bike throttle
Dimmer switch
Light switch
Water tap
Music on a CD
Music on a tape

Analog and Digital Signals

A security floodlight switches on when you approach.

It has an **analogue input** (how much infra red it sees from you), and produces a **digital output** (the floodlight is either on or off).

We could call it an "**analog to digital converter**".

Analog and Digital Signals

- The problem with analog signals is **noise** – hiss on the sound and speckly dots on the picture.
- When we send a signal over a long distance, the signal gets weaker, so we need to boost (amplify) it.
- The problem is that we end up boosting the noise as well.

Analog and Digital Signals

If we convert the signal into digital form, then send it, it still gets weaker and noise still creeps in.

Analog and Digital Signals

Example: if you have a bad photocopy of a piece of text, and you photocopy that, you’ll get a worse photocopy.

But if you **read the text yourself**, the “software” in your brain can “reconstruct” the text, because **you know what the letter shapes are supposed to be** even though they’re blurred.

Analog and Digital Signals

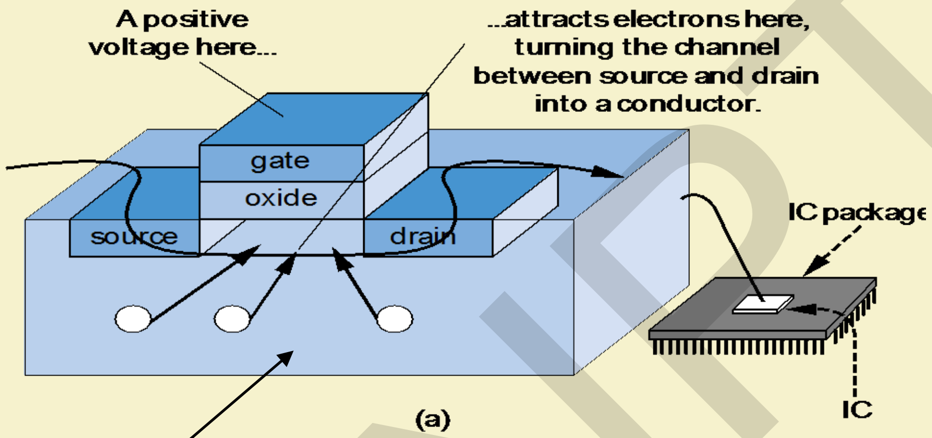
<u>Summary</u>:

Analog signals suffer from noise, but don't need such complex equipment.

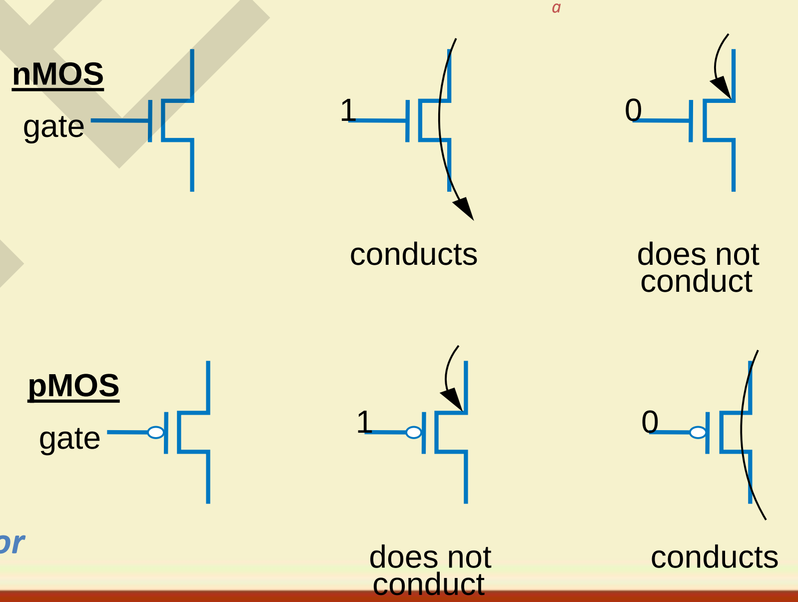
Digital signals need fast, clever electronics, but we can get rid of any noise.

The CMOS Transistor

- CMOS transistor
  - Basic switch in modern ICs



Silicon -- not quite a conductor or insulator: ***Semiconductor***



Digital Circuits

Number System

Santanu Chattopadhyay
Electronics and Electrical Communication Engineering

Common Number Systems

System	Base	Symbols	Used by humans?	Used in computers?
Decimal	10	0, 1, … 9	Yes	No
Binary	2	0, 1	No	Yes
Octal	8	0, 1, … 7	No	No
Hexa-decimal	16	0, 1, … 9, A, B, … F	No	No

Quantities/Counting (1 of 3)

Decimal	Binary	Octal	Hexa-decimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7

Quantities/Counting (2 of 3)

Decimal	Binary	Octal	Hexa-decimal
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

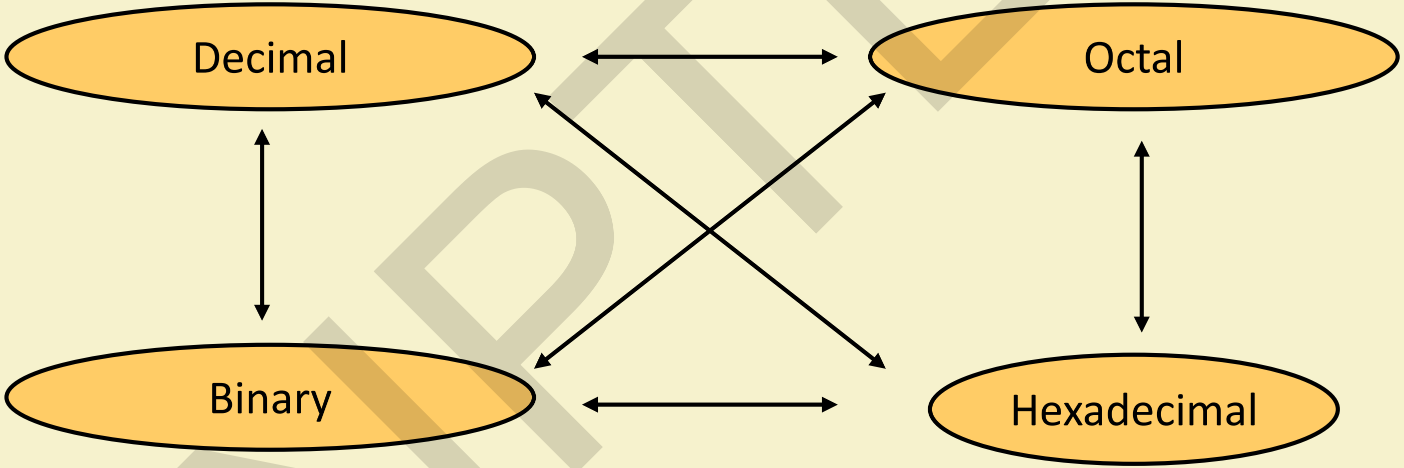
Quantities/Counting (3 of 3)

Decimal	Binary	Octal	Hexa-decimal
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14
21	10101	25	15
22	10110	26	16
23	10111	27	17

Etc.

Conversion Among Bases

- The possibilities:

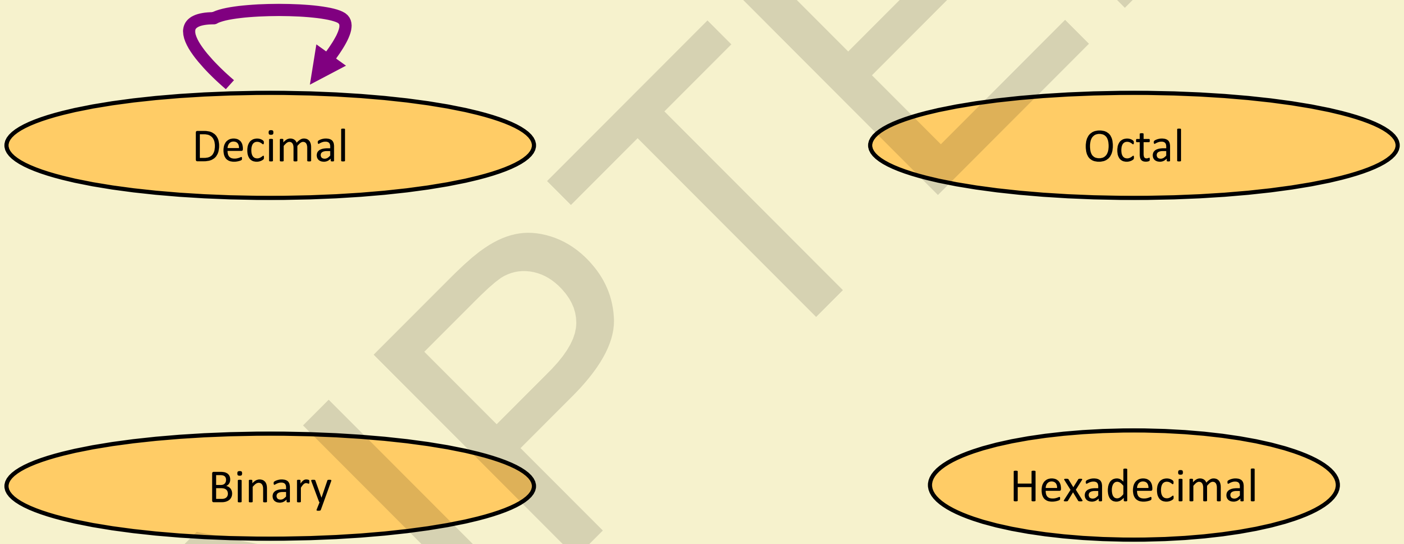


Quick Example

$$25_{10} = 11001_2 = 31_8 = 19_{16}$$

Base

Decimal to Decimal (just for fun)

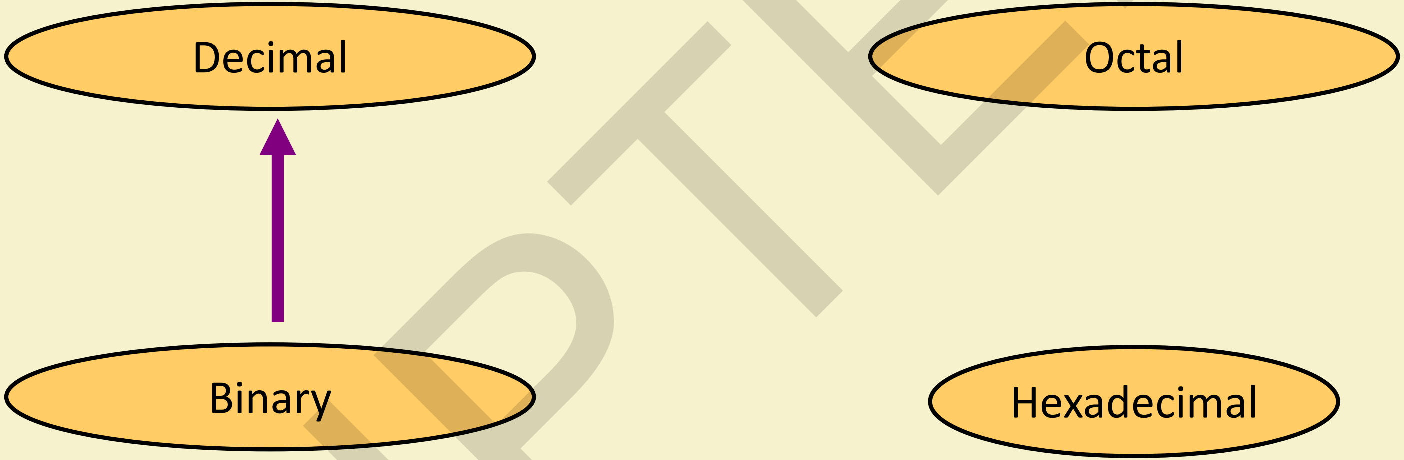


Weight

$$125_{10} \Rightarrow \begin{array}{rcr} 5 \times 10^0 & = & 5 \\ 2 \times 10^1 & = & 20 \\ 1 \times 10^2 & = & 100 \\ \hline & & 125 \end{array}$$

Base

Binary to Decimal



Binary to Decimal

- Technique
  - Multiply each bit by 2^n, where n is the "weight" of the bit
  - The weight is the position of the bit, starting from 0 on the right
  - Add the results

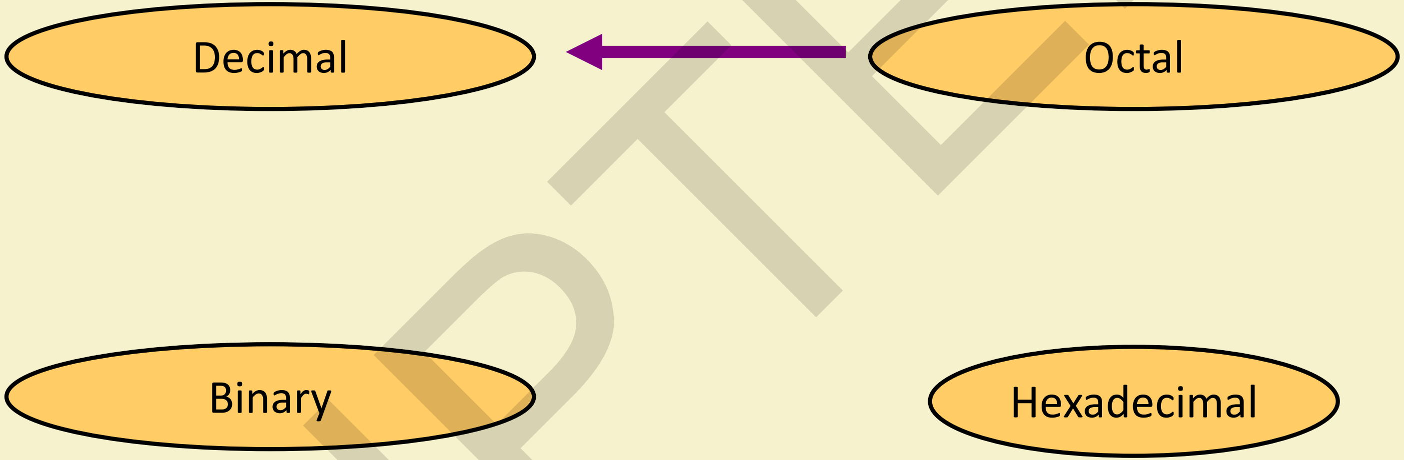
Example

Bit "0"

101011_2 =>

$$
\begin{array}{lllll}
1 & \times & 2^0 & = & 1 \\
1 & \times & 2^1 & = & 2 \\
0 & \times & 2^2 & = & 0 \\
1 & \times & 2^3 & = & 8 \\
0 & \times & 2^4 & = & 0 \\
1 & \times & 2^5 & = & \underline{32} \\
 & & & & 43_{10}
\end{array}
$$

Octal to Decimal



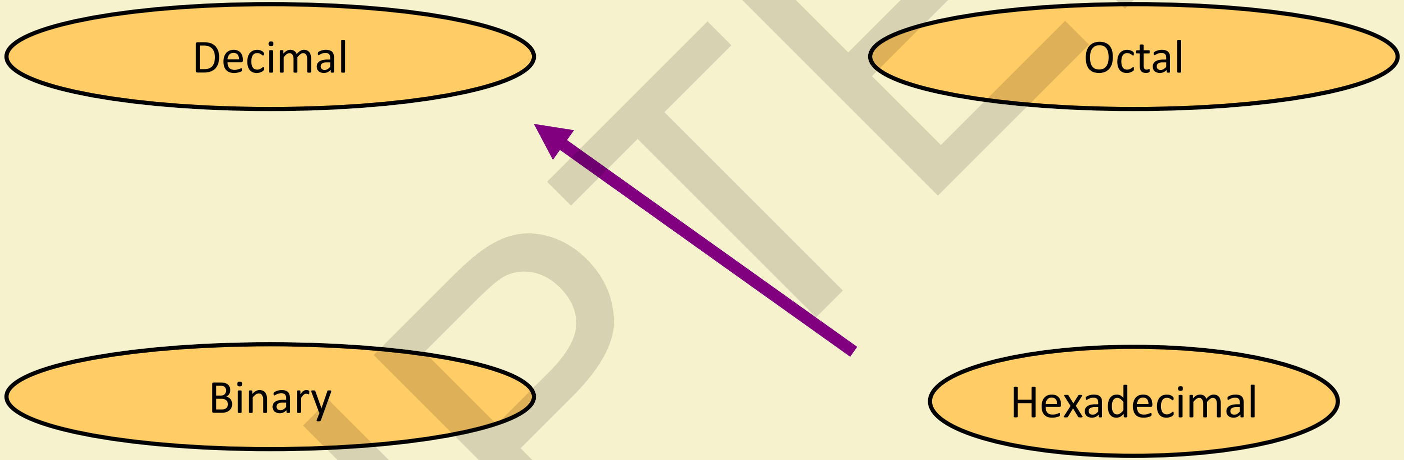
Octal to Decimal

- Technique
  - Multiply each digit by 8^n, where n is the "weight" of the digit
  - The weight is the position of the digit, starting from 0 on the right
  - Add the results

Example

$$
\begin{array}{lrcr}
724_8 \Rightarrow & 4 \times 8^0 & = & 4 \\
& 2 \times 8^1 & = & 16 \\
& 7 \times 8^2 & = & 448 \\
\hline
& & & 468_{10}
\end{array}
$$

Hexadecimal to Decimal



Hexadecimal to Decimal

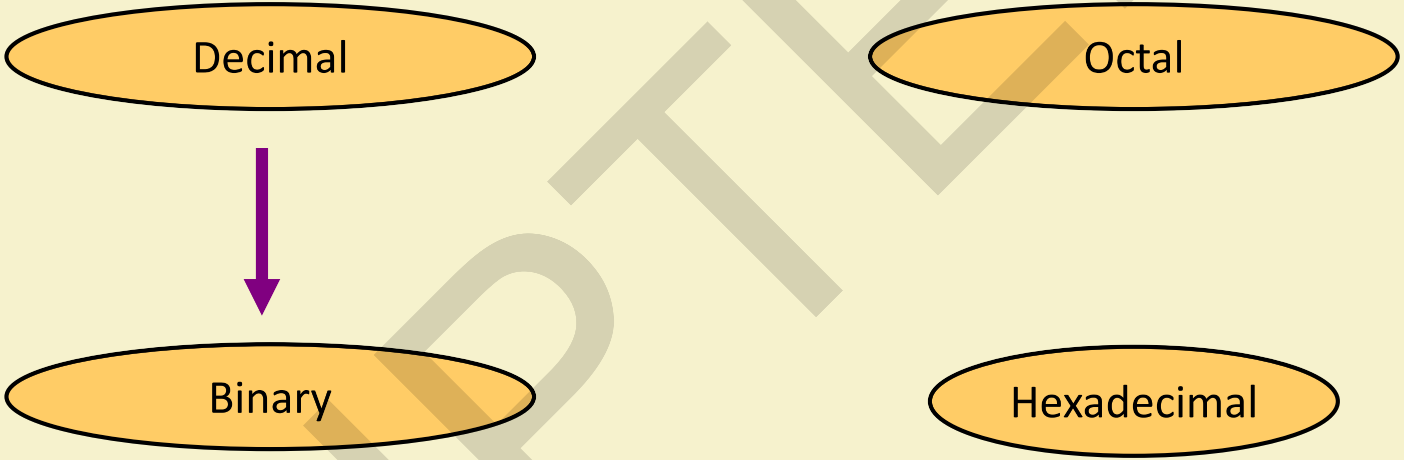
- Technique
  - Multiply each digit by 16^n, where n is the "weight" of the digit
  - The weight is the position of the digit, starting from 0 on the right
  - Add the results

Example

ABC_{16} =>

$$
\begin{array}{l}
C \times 16^0 = 12 \times 1 = 12 \\
B \times 16^1 = 11 \times 16 = 176 \\
A \times 16^2 = 10 \times 256 = 2560 \\
\hline
2748_{10}
\end{array}
$$

Decimal to Binary



Decimal to Binary

- Technique
  - Divide by two, keep track of the remainder
  - First remainder is bit 0 (LSB, least-significant bit)
  - Second remainder is bit 1
  - Etc.

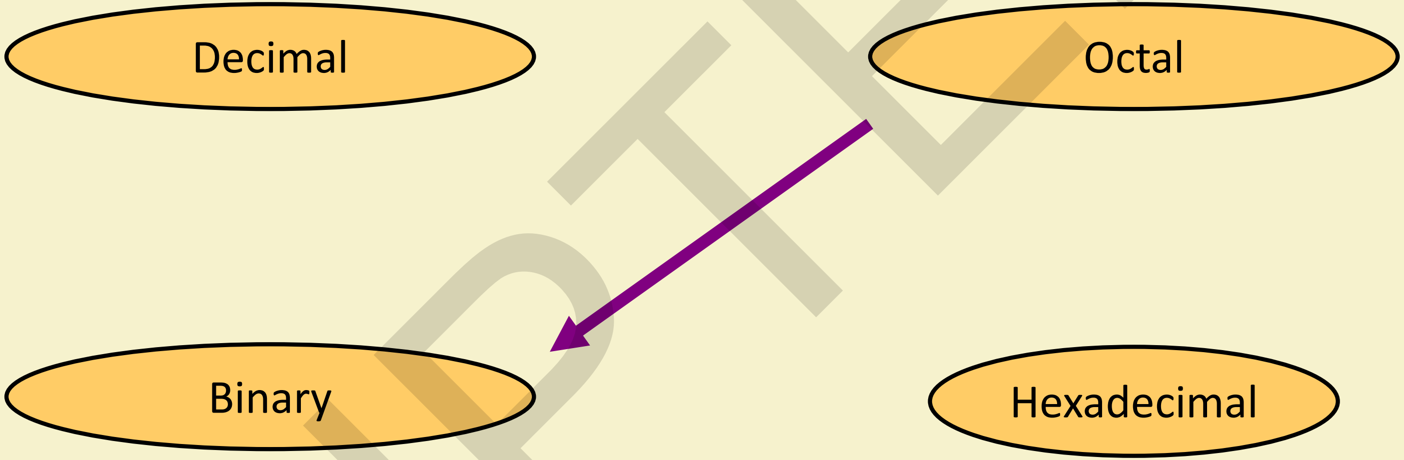
Example

$125_{10} = ?_2$

2	125	
2	62	1
2	31	0
2	15	1
2	7	1
2	3	1
2	1	1
	0	1

$125_{10} = 1111101_2$

Octal to Binary



Octal to Binary

- Technique
  - Convert each octal digit to a 3-bit equivalent binary representation

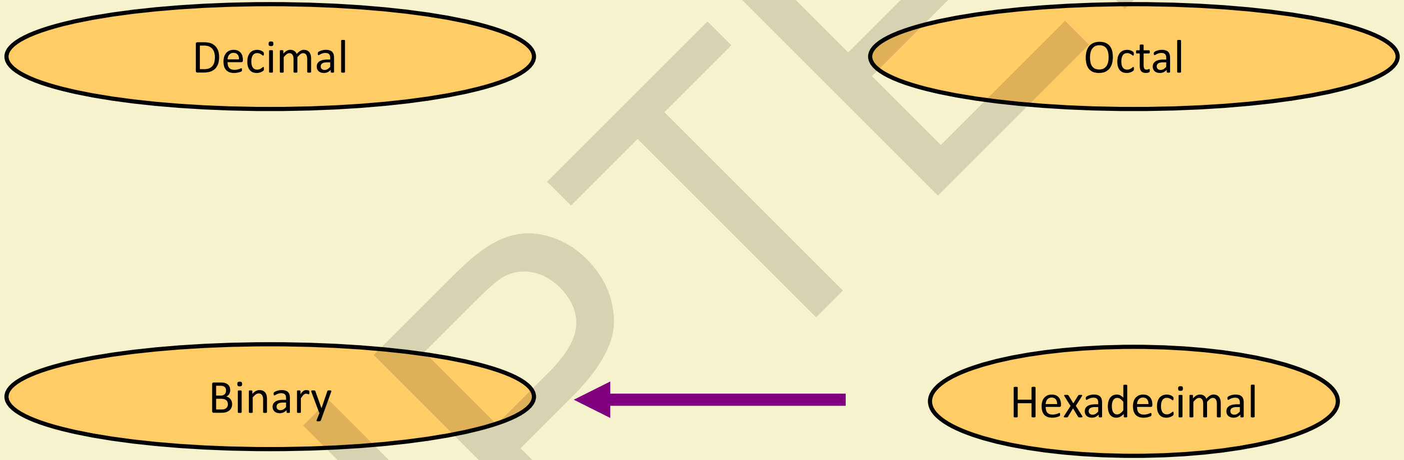
Example

$705_8 = ?_2$

$$\begin{array}{ccc} 7 & 0 & 5 \\ \downarrow & \downarrow & \downarrow \\ 111 & 000 & 101 \end{array}$$

$705_8 = 111000101_2$

Hexadecimal to Binary



Hexadecimal to Binary

- Technique
  - Convert each hexadecimal digit to a 4-bit equivalent binary representation

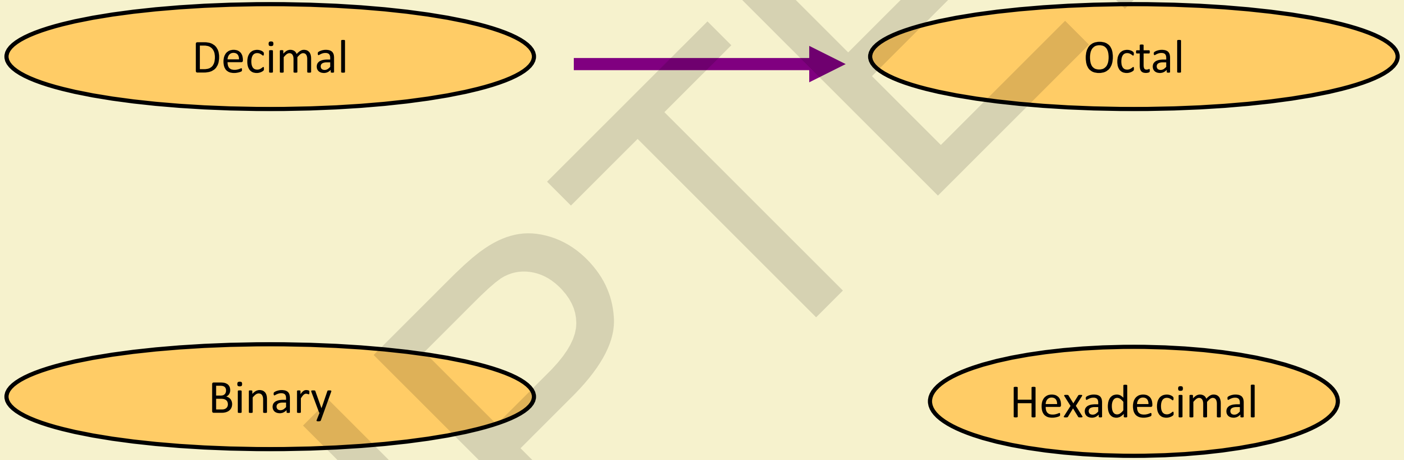
Example

$10AF_{16} = ?_2$

1	0	A	F
0001	0000	1010	1111

$10AF_{16} = 0001000010101111_2$

Decimal to Octal



Decimal to Octal

- Technique
  - Divide by 8
  - Keep track of the remainder

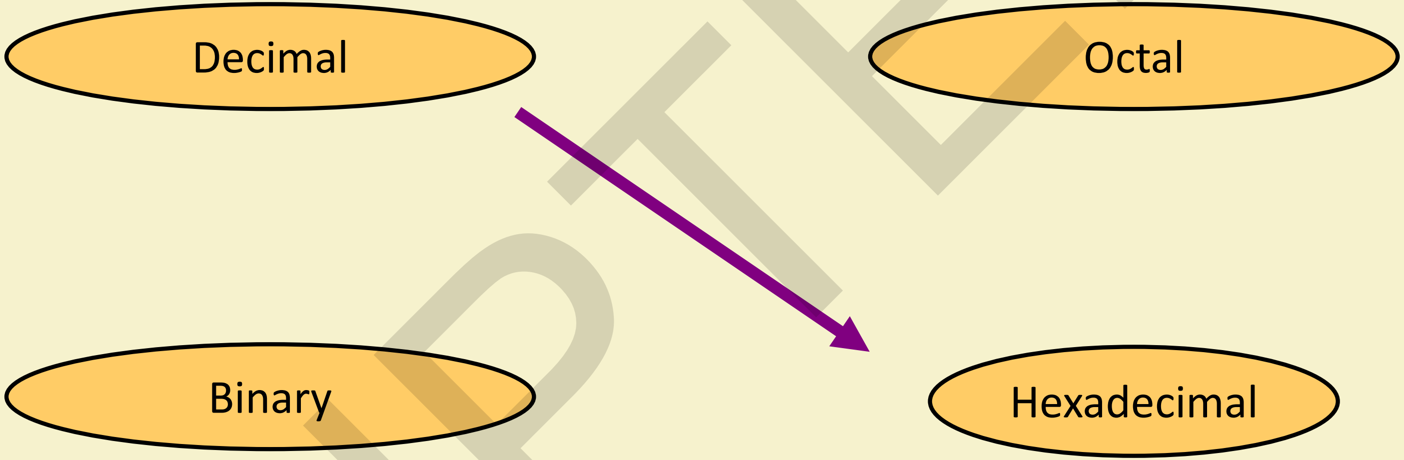
Example

$1234_{10} = ?_8$

8	1234	
8	154	2
8	19	2
8	2	3
	0	2

$1234_{10} = 2322_8$

Decimal to Hexadecimal



Decimal to Hexadecimal

- Technique
  - Divide by 16
  - Keep track of the remainder

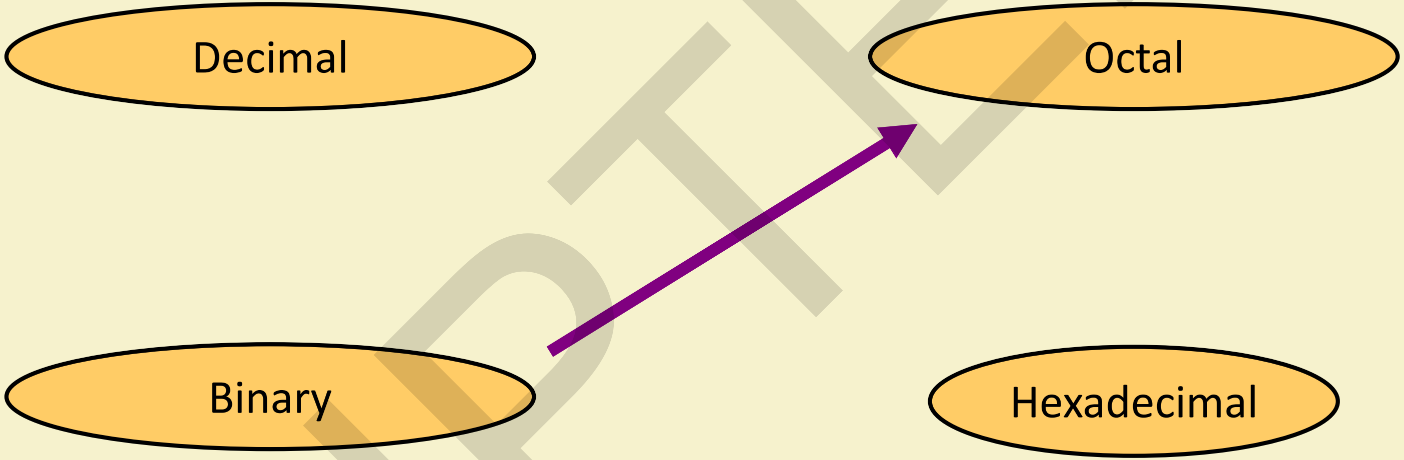
Example

$1234_{10} = ?_{16}$

16	1234	
16	77	2
16	4	13 = D
	0	4

$1234_{10} = 4D2_{16}$

Binary to Octal



Binary to Octal

- Technique
  - Group bits in threes, starting on right
  - Convert to octal digits

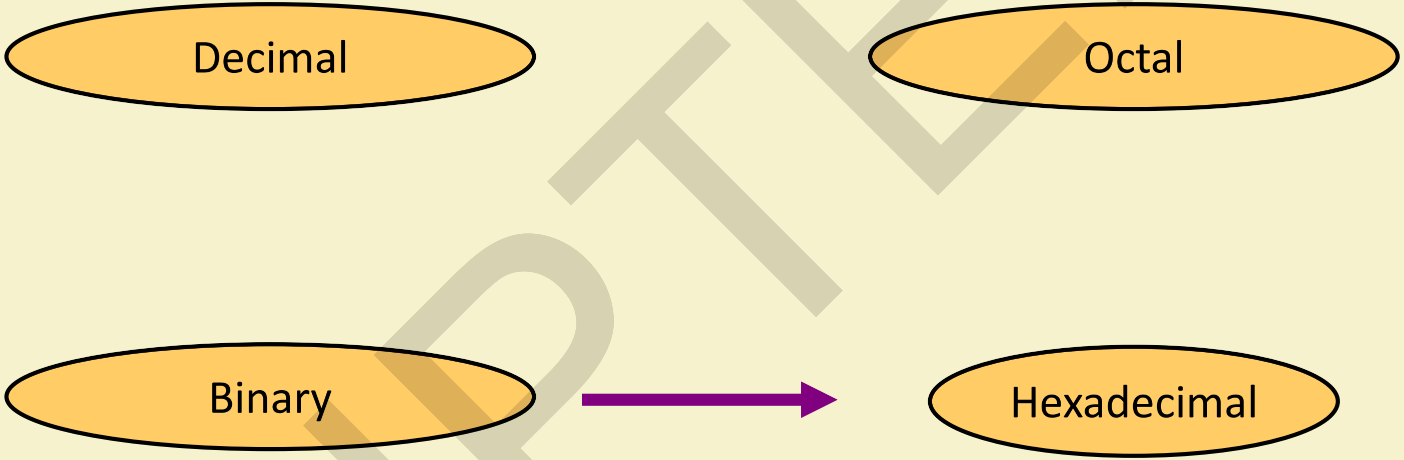
Example

$1011010111_2 = ?_8$

$$
\begin{array}{cccc}
1 & 011 & 010 & 111 \\
\downarrow & \downarrow & \downarrow & \downarrow \\
1 & 3 & 2 & 7
\end{array}
$$

$1011010111_2 = 1327_8$

Binary to Hexadecimal



Binary to Hexadecimal

- Technique
  - Group bits in fours, starting on right
  - Convert to hexadecimal digits

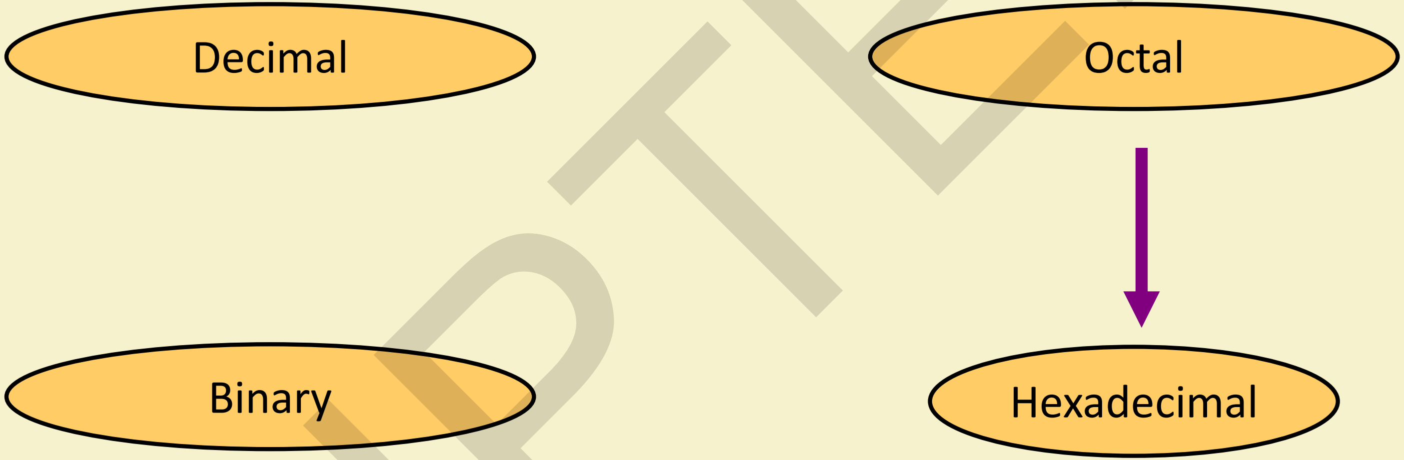
Example

$1010111011_2 = ?_{16}$

$$\begin{array}{ccc} 10 & 1011 & 1011 \\ \downarrow & \downarrow & \downarrow \\ 2 & B & B \end{array}$$

$1010111011_2 = 2BB_{16}$

Octal to Hexadecimal



Octal to Hexadecimal

- Technique
  – Use binary as an intermediary

Example

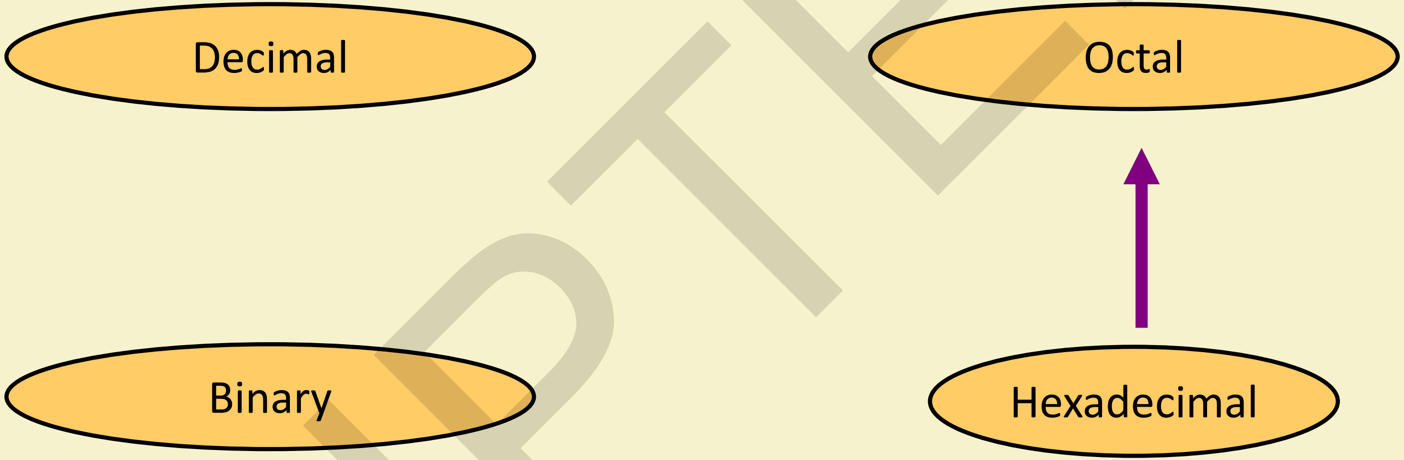
$1076_8 = ?_{16}$

1	0	7	6
↓	↓	↓	↓
001	000	111	110

2 3 E

$1076_8 = 23E_{16}$

Hexadecimal to Octal



Hexadecimal to Octal

- Technique
  - Use binary as an intermediary

Example

$1F0C_{16} = ?_8$

1	F	0	C
0001	1111	0000	1100

$0 \mid 001 \mid 111 \mid 100 \mid 001 \mid 100$

1 7 4 1 4

$1F0C_{16} = 17414_8$

Exercise – Convert ...

Decimal	Binary	Octal	Hexa-decimal
33			
	1110101		
		703	
			1AF

Exercise – Convert …

Answer

Decimal	Binary	Octal	Hexa-decimal
33	100001	41	21
117	1110101	165	75
451	111000011	703	1C3
431	110101111	657	1AF

Common Powers (1 of 2)

- Base 10

Power	Preface	Symbol	Value
10^{-12}	pico	p	.000000000001
10^{-9}	nano	n	.000000001
10^{-6}	micro	µ	.000001
10^{-3}	milli	m	.001
10^{3}	kilo	k	1000
10^{6}	mega	M	1000000
10^{9}	giga	G	1000000000
10^{12}	tera	T	1000000000000

Common Powers (2 of 2)

- Base 2

Power	Preface	Symbol	Value
2^{10}	kilo	k	1024
2^{20}	mega	M	1048576
2^{30}	Giga	G	1073741824

- What is the value of “k”, “M”, and “G”?
- In computing, particularly w.r.t. <u>memory</u>, the base-2 interpretation generally applies

Review – multiplying powers

- For common bases, add powers

$$a^b \times a^c = a^{b+c}$$

$$2^6 \times 2^{10} = 2^{16} = 65{,}536$$

or…

$$2^6 \times 2^{10} = 64 \times 2^{10} = 64\text{k}$$

Binary Addition (1 of 2)

- Two 1-bit values

A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	10

"two"

Binary Addition (2 of 2)

- Two *n*-bit values
  - Add individual bits
  - Propagate carries
  - E.g.,

```
  1   1
   10101       21
 + 11001     + 25
 -------     ----
  101110       46
```

Multiplication (1 of 3)

- Decimal (just for fun)

```
    35
 x 105
 -----
   175
  000
 35
 -----
  3675
```

Multiplication (2 of 3)

- Binary, two 1-bit values

A	B	$A \times B$
0	0	0
0	1	0
1	0	0
1	1	1

Multiplication (3 of 3)

- Binary, two *n*-bit values
  - As with decimal values
  - E.g.,

```
       1110
     x 1011
     ------
       1110
      1110
     0000
    1110
   --------
   10011010
```

Fractions

- Decimal to decimal (just for fun)

```
3.14 =>       4 x 10^-2 = 0.04
              1 x 10^-1 = 0.1
              3 x 10^0  = 3
                        ------
                          3.14
```

$$3.14 \Rightarrow \begin{array}{l} 4 \times 10^{-2} = 0.04 \\ 1 \times 10^{-1} = 0.1 \\ 3 \times 10^{0} = \underline{3\qquad} \\ \qquad\qquad\ \ 3.14 \end{array}$$

Fractions

- Binary to decimal

10.1011 =>

$$
\begin{array}{l}
1 \times 2^{-4} = 0.0625 \\
1 \times 2^{-3} = 0.125 \\
0 \times 2^{-2} = 0.0 \\
1 \times 2^{-1} = 0.5 \\
0 \times 2^{0} = 0.0 \\
1 \times 2^{1} = \underline{2.0} \\
\qquad\qquad 2.6875
\end{array}
$$

Fractions

- Decimal to binary

```
3.14579

11.001001...
```

```
 .14579
x      2
0.29158
x      2
0.58316
x      2
1.16632
x      2
0.33264
x      2
0.66528
x      2
1.33056
etc.
```

Exercise – Convert …

Decimal	Binary	Octal	Hexa-decimal
29.8	11101.110011…	35.63…	1D.CC…
5.8125	101.1101	5.64	5.D
3.109375	11.000111	3.07	3.1C
12.5078125	1100.10000010	14.404	C.82

Negative Numbers

1. Sign and Magnitude Representation
2. 1’s Complement Representation
3. 2’s Complement Representation

Goal of negative number systems

- Signed system: Simple. Just flip the sign bit
    - 0 = positive
    - 1 = negative
- One’s complement: Replace subtraction with addition
    - Easy to derive (Just flip every bit)
- Two’s complement: Replace subtraction with addition
    - Addition of one’s complement and one produces the two’s complement.

Given a positive integer x, we represent -x

- 1's complement:
  - Formula: $2^n - 1 - x$
    - i.e. n=4, $2^4 - 1 - x = 15 - x$
    - In binary: $(1\ 1\ 1\ 1) - (b_3\ b_2\ b_1\ b_0)$
    - Just flip all the bits.
- 2's complement:
  - Formula: $2^n - x$
    - i.e. n=4, $2^4 - x = 16 - x$
    - In binary: $(1\ 0\ 0\ 0\ 0) - (0\ b_3\ b_2\ b_1\ b_0)$
    - Just flip all the bits and add 1.

Definitions:
4-Bit Example

id	b_3	b_2	b_1	b_0	Signed	One's	Two's
0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	1
2	0	0	1	0	2	2	2
3	0	0	1	1	3	3	3
4	0	1	0	0	4	4	4
5	0	1	0	1	5	5	5
6	0	1	1	0	6	6	6
7	0	1	1	1	7	7	7
8	1	0	0	0	-0	-7	-8
9	1	0	0	1	-1	-6	-7
10	1	0	1	0	-2	-5	-6
11	1	0	1	1	-3	-4	-5
12	1	1	0	0	-4	-3	-4
13	1	1	0	1	-5	-2	-3
14	1	1	1	0	-6	-1	-2
15	1	1	1	1	-7	-0	-1

Given n-bits, what is the range of my numbers in each system?

- 3 bits:
  - Signed: -3 , 3
  - 1's: -3 , 3
  - 2's: -4 , 3
- 6 bits
  - Signed: -31, 31
  - 1's: -31, 31
  - 2's: -32, 31
- 5 bits:
  - Signed: -15, 15
  - 1's: -15, 15
  - 2's: -16, 15
- Given 8 bits
  - Signed: -127, 127
  - 1's: -127, 127
  - 2's: -128, 127

Formula for calculating the range →

Signed & 1's: $-(2^{n-1} - 1)$, $(2^{n-1} - 1)$

2's: -2^{n-1} , $(2^{n-1} - 1)$

Arithmetic Operations:

Derivation of 1's Complement

Theorem 1: For 1's complement, given a positive number (x_{n-1} , x_{n-2} , … , x_0), the negative number is $(\bar{x}_{n-1}, \bar{x}_{n-2}, \ldots, \bar{x}_0)$ where $\bar{x} = 1 - x$

Proof:

(i). 2^n-1 in binary is an n bit vector (1,1, …, 1)

(ii). 2^n-1-x in binary is (1,1, …,1) – (x_{n-1},x_{n-2}, …, x_0).

The result is

$$(\bar{x}_{n-1}, \bar{x}_{n-2}, \ldots, \bar{x}_0)$$

Arithmetic Operations: Derivation of 2's Complement

Theorem 2: For 2's complement, given a positive integer x, the negative number is the sum of its 1's complement and 1.

Proof: $2^n - x = 2^n - 1 - x + 1$. Thus, the 2's complement is

$$(\bar{x}_{n-1}, \bar{x}_{n-2}, \cdots, \bar{x}_0) + (0, 0, \ldots, 1)$$

Ex: x = 9 (01001)

1's -9 (10110)

31 – 9 = 22

2's -9 (10111)

32 – 9 = 23

Ex: x = 13 (01101)

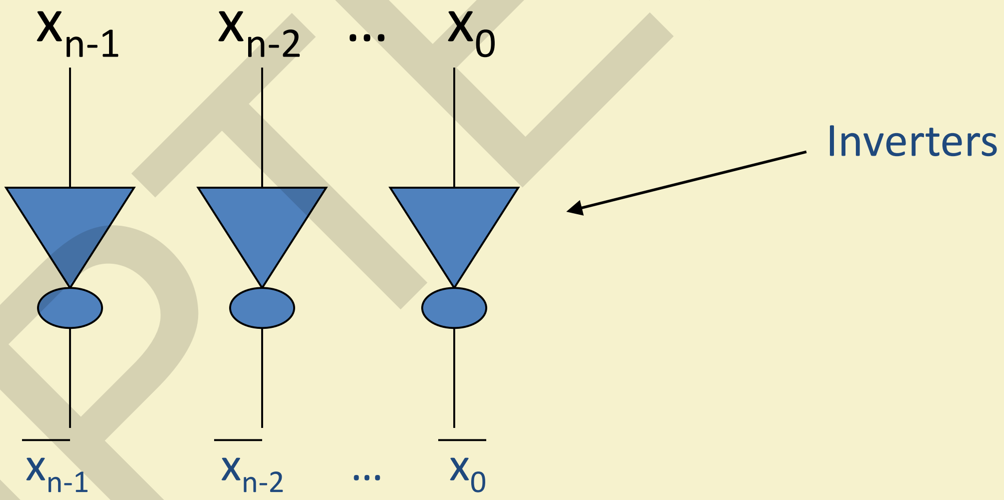
1's -13 (10010)

31-13=18

2's -13 (10011)

32-13=19

One's Complement Hardware:



Arithmetic Operations: 2's Complement

Input: two positive integers x & y,

1. We represent the operands in two's complement.
2. We sum up the two operands and ignore bit n.
3. The result is the solution in two's complement.

Arithmetic	2's complement
$x + y$	$x + y$
$x - y$	$x + (2^n - y) = 2^n+(x-y)$
$-x + y$	$(2^n - x) + y =2^n+(-x+y)$
$-x - y$	$(2^n - x) + (2^n - y) = 2^n+2^n-x-y$

Arithmetic Operations: Example: 4 – 3 = 1

$4_{10} = 0100_2$

$3_{10} = 0011_2$ $\quad -3_{10} \rightarrow 1101_2$

$$\begin{array}{r} 0100 \\ \underline{+\ 1101} \\ 10001 \end{array} \rightarrow 1 \text{ (after discarding extra bit)}$$

We discard the extra 1 at the left which is 2^n from 2's complement of -3. Note that bit b_{n-1} is 0. Thus, the result is positive.

Arithmetic Operations: Example: -4 +3 = -1

$4_{10} = 0100_2$ $-4_{10} \rightarrow$ Using two's comp. $\rightarrow$ $1011 + 1 = 1100_2$

(Invert bits)

$3_{10} = 0011_2$

$$\begin{array}{r} 1100 \\ +\ 0011 \\ \hline 1111 \end{array}$$

$\rightarrow$ Using two's comp. $\rightarrow$ 0000 + 1 = 1, so our answer is -1

If left-most bit is 1, it means that we have a negative number.

Arithmetic Operations: 1's Complement

Input: two positive integers x & y,

1. We represent the operands in one's complement.
2. We sum up the two operands.
3. We subtract 2^n-1 if there is carry out at left.
4. The result is the solution in one's complement.

Arithmetic	1's complement
$x + y$	$x + y$
$x - y$	$x + (2^n - 1 - y) = 2^n - 1 + (x - y)$
$-x + y$	$(2^n - 1 - x) + y = 2^n - 1 + (-x + y)$
$-x - y$	$(2^n - 1 - x) + (2^n - 1 - y) = 2^n - 1 + (2^n - 1 - x - y)$

Arithmetic Operations: Example: 4 – 3 = 1

$4_{10} = 0100_2$

$3_{10} = 0011_2$ $-3_{10} \rightarrow 1100_2$ in one's complement

0100 (4 in decimal)

+ 1100 (12 in decimal or 15-3)

1,0000 (16 in decimal or 15+1)

0001(after subtracting 2^n-1)

We discard the extra 1 at the left which is 2^n and add one at the first bit.

Arithmetic Operations: Example: -4 +3 = -1

$4_{10} = 0100_2$ $-4_{10} \rightarrow$ Using one's comp. $\rightarrow 1011_2$

(Invert bits)

$3_{10} = 0011_2$

1011 (11 in decimal or 15-4)

+ 0011 (3 in decimal)

1110 (14 in decimal or 15-1)

If the left-most bit is 1, it means that we have a negative number.