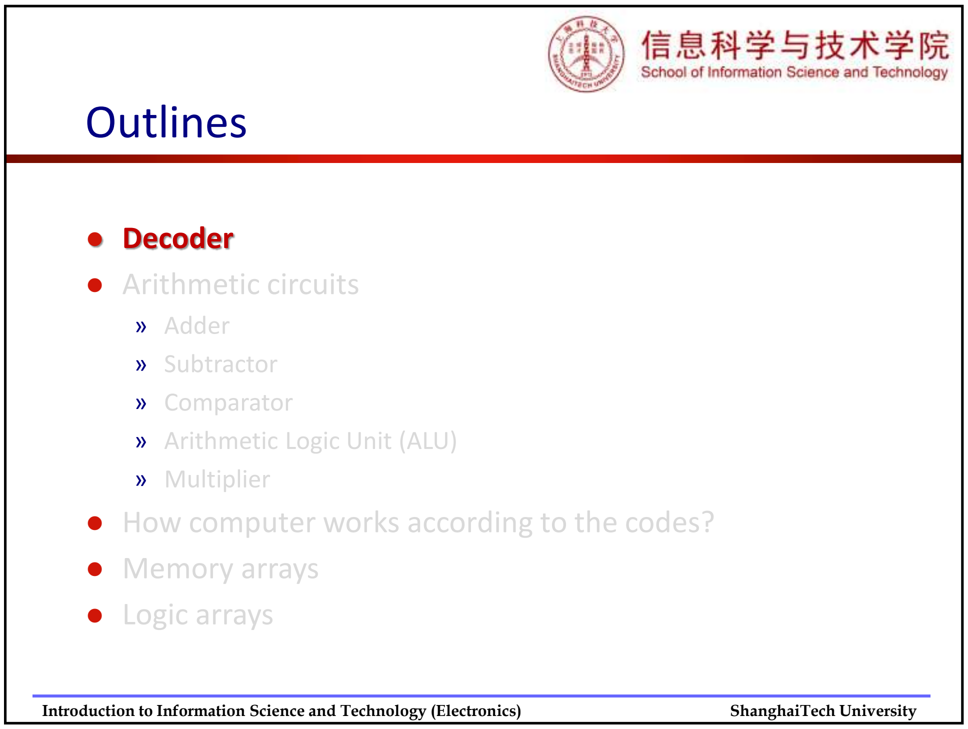
Lecture 5

Digital Building Blocks

Junrui Liang
ShanghaiTech University

Outlines

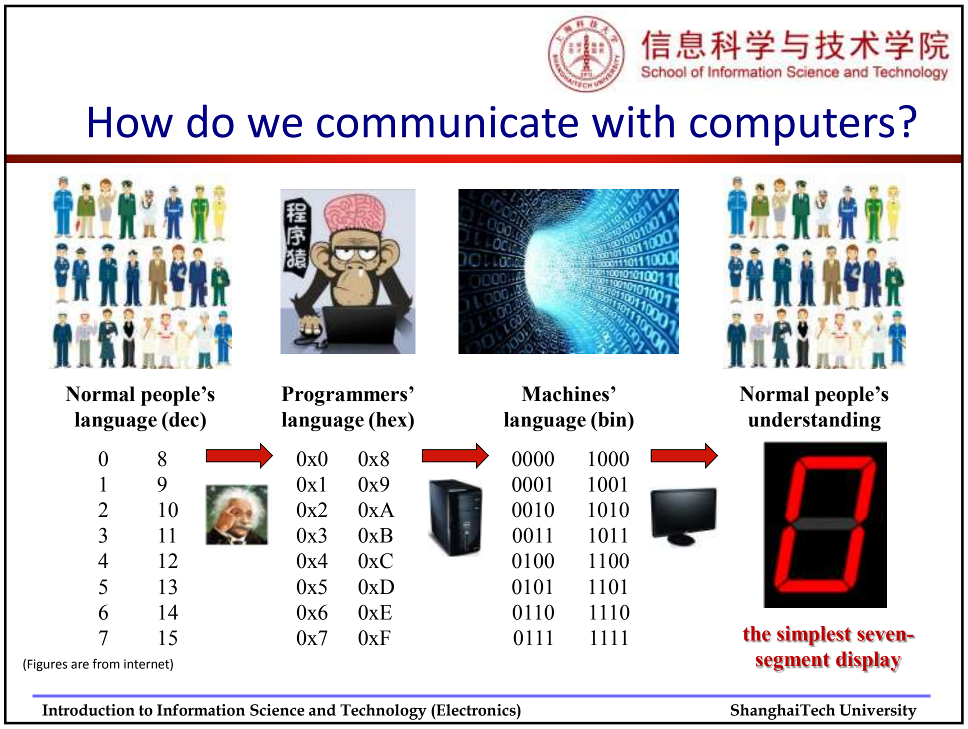
- Decoder
- Arithmetic circuits
  - Adder
  - Subtractor
  - Comparator
  - Arithmetic Logic Unit (ALU)
  - Multiplier
- How computer works according to the codes?
- Memory arrays
- Logic arrays



信息科学与技术学院
School of Information Science and Technology

Outlines

- **Decoder**
- Arithmetic circuits
  - » Adder
  - » Subtractor
  - » Comparator
  - » Arithmetic Logic Unit (ALU)
  - » Multiplier
- How computer works according to the codes?
- Memory arrays
- Logic arrays



信息科学与技术学院
School of Information Science and Technology

How do we communicate with computers?

Normal people's language (dec)		Programmers' language (hex)		Machines' language (bin)	
0	8	0x0	0x8	0000	1000
1	9	0x1	0x9	0001	1001
2	10	0x2	0xA	0010	1010
3	11	0x3	0xB	0011	1011
4	12	0x4	0xC	0100	1100
5	13	0x5	0xD	0101	1101
6	14	0x6	0xE	0110	1110
7	15	0x7	0xF	0111	1111

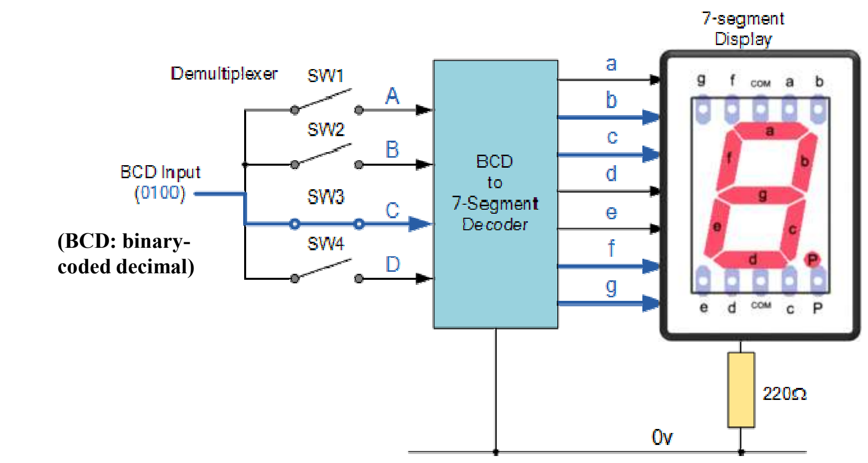
Normal people's understanding

the simplest seven-segment display

(Figures are from internet)

信息科学与技术学院
School of Information Science and Technology

Seven Segment Decoder



(BCD: binary-coded decimal)

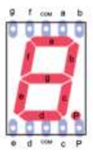
(Figures are from internet)

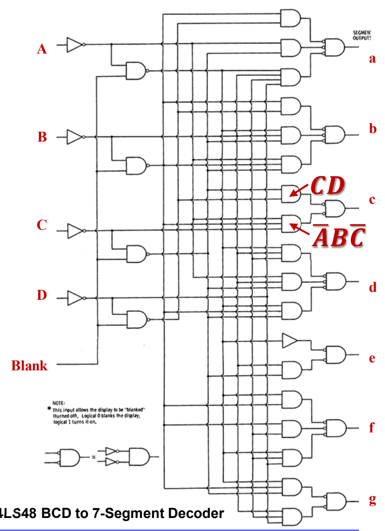
信息科学与技术学院
School of Information Science and Technology

The BCD to Seven-Segment Decoder

- Truth table

DESIMAL	D	C	B	A	a	b	c	d	e	f	g	7-LED
0	0	0	0	0	0	0	0	0	0	0	1	
1	0	0	0	1	1	0	0	1	1	1	1	
2	0	0	1	0	0	0	1	0	0	1	0	
3	0	0	1	1	0	0	0	0	1	1	0	
4	0	1	0	0	1	0	0	1	1	0	0	
5	0	1	0	1	0	1	0	0	1	0	0	
6	0	1	1	0	1	1	0	0	0	0	0	
7	0	1	1	1	0	0	0	1	1	1	1	
8	1	0	0	0	0	0	0	0	0	0	0	
9	1	0	0	1	0	0	0	1	1	0	0	
10	1	0	1	0	1	1	1	0	0	1	0	
11	1	0	1	1	1	1	0	0	1	1	0	
12	1	1	0	0	1	0	1	1	1	0	0	
13	1	1	0	1	0	1	1	0	1	0	0	
14	1	1	1	0	1	1	1	0	0	0	0	
15	1	1	1	1	1	1	1	1	1	1	1	





74LS48 BCD to 7-Segment Decoder

Example: $c = CD + \overline{A}B\overline{C} = \overline{\overline{CD}\;\overline{\overline{A}B\overline{C}}}$

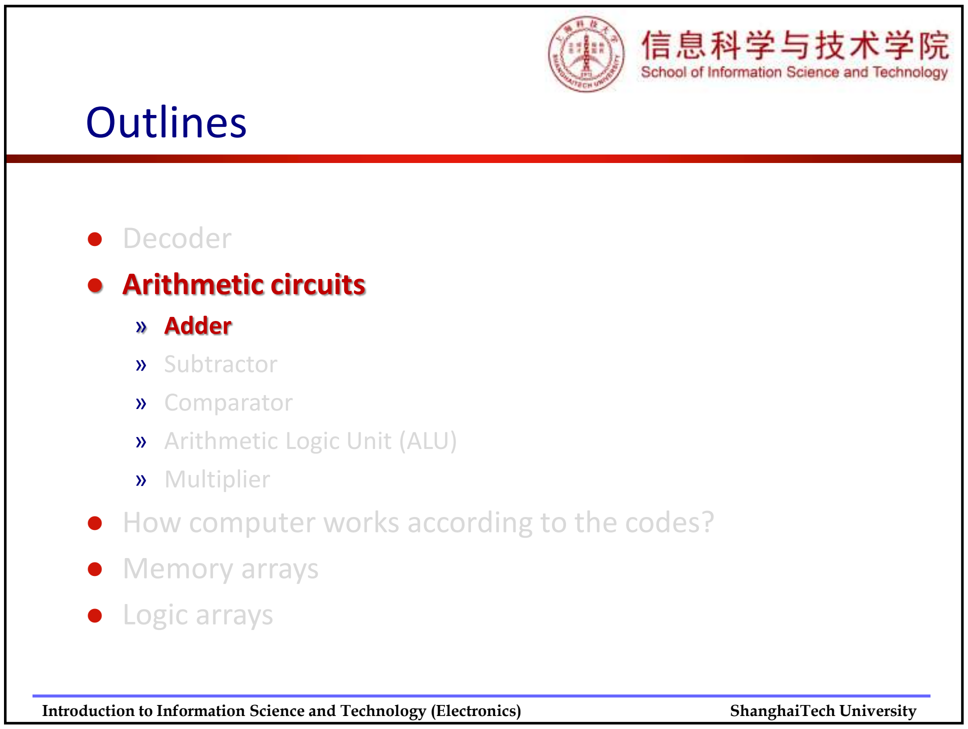
(Figures are from internet)

Outlines

- Decoder
- **Arithmetic circuits**
  - » Adder
  - » Subtractor
  - » Comparator
  - » Arithmetic Logic Unit (ALU)
  - » Multiplier
- How computer works according to the codes?
- Memory arrays
- Logic arrays

Arithmetic Circuits

- The ***central building blocks*** of computers
- Perform many arithmetic functions:
  - » Addition 加法
  - » Subtraction 减法
  - » Comparisons 比较
  - » Shifts 移位
  - » Multiplication 成分
  - » Division 除法

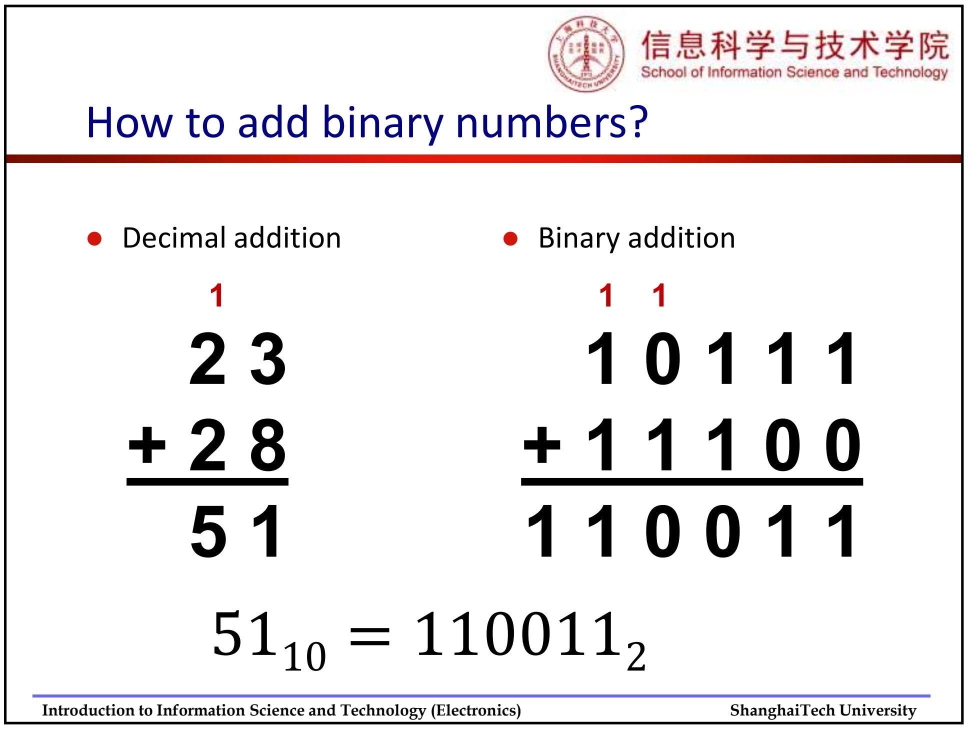


信息科学与技术学院
School of Information Science and Technology

Outlines

- Decoder
- **Arithmetic circuits**
  - » **Adder**
  - » Subtractor
  - » Comparator
  - » Arithmetic Logic Unit (ALU)
  - » Multiplier
- How computer works according to the codes?
- Memory arrays
- Logic arrays

Introduction to Information Science and Technology (Electronics) ShanghaiTech University



信息科学与技术学院
School of Information Science and Technology

How to add binary numbers?

- Decimal addition

$$\begin{array}{r} {\scriptstyle 1} \\ 2\,3 \\ +\,2\,8 \\ \hline 5\,1 \end{array}$$

- Binary addition

$$\begin{array}{r} {\scriptstyle 11} \\ 1\,0\,1\,1\,1 \\ +\,1\,1\,1\,0\,0 \\ \hline 1\,1\,0\,0\,1\,1 \end{array}$$

$$51_{10} = 110011_2$$

Introduction to Information Science and Technology (Electronics) ShanghaiTech University

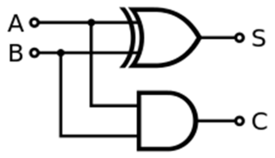
Half Adder

Inputs		Outputs	
A	*B*	C_{out}	*S*
0	0	0	0
1	0	0	1
0	1	0	1
1	1	1	0

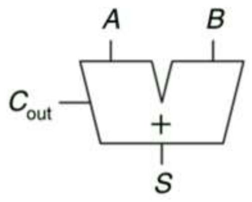
The sum: $S = A \oplus B$

The carry (out): $C = AB$

Logic diagram



Symbol



Full Adder

Inputs			Outputs	
A	*B*	C_{in}	C_{out}	*S*
0	0	0	0	0
1	0	0	0	1
0	1	0	0	1
1	1	0	1	0
0	0	1	0	1
1	0	1	1	0
0	1	1	1	0
1	1	1	1	1

$$S = (A \oplus B)\overline{C}_{in} + (\overline{A \oplus B})C_{in}$$

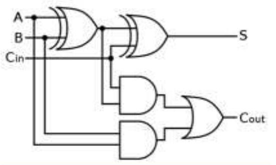
$$= A \oplus B \oplus C_{in}$$

$$C_{out} = AB\overline{C}_{in} + A\overline{B}C_{in} + \overline{A}BC_{in} + ABC_{in}$$

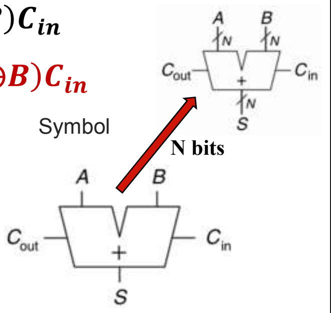
$$= AB + (A + B)C_{in}$$

$$\text{or } AB + (A \oplus B)C_{in}$$

Logic diagram



Symbol



信息科学与技术学院
School of Information Science and Technology

Outlines

- Decoder
- **Arithmetic circuits**
  - » Adder
  - » **Subtractor**
  - » Comparator
  - » Arithmetic Logic Unit (ALU)
  - » Multiplier
- How computer works according to the codes?
- Memory arrays
- Logic arrays

信息科学与技术学院
School of Information Science and Technology

How computer reads negative number

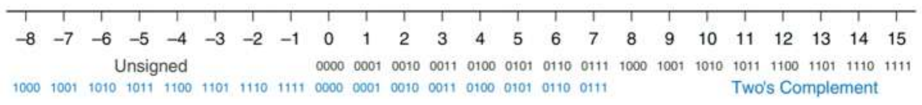
- Unsigned and signed binary numbers
  - » 8 bit unsigned number
    0 to 255, i.e., 0 ~ 0xFF
  - » 8 bit signed number
    -128 to 127, i.e., ?? to 0x7F

How does the computer knows the minus sign "–"?

- Two's Complement Numbers (2的补数)
  1. Invert all bits
  2. Add "1" to the last significant bit

Example for four bits number:

$$5_{10} = 0101_2$$
$$(-5_{10}) = 1010_2 + 1$$
$$= 1011_2$$



Binary Subtraction

- Decimal subtraction

$$\begin{array}{r} 15 \\ -\ \ 5 \\ \hline 10 \end{array}$$

$$\begin{array}{r} 15 \\ +(-5) \\ \hline 10 \end{array}$$

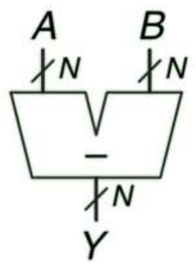
- Binary subtraction

$$\begin{array}{rcccccccc} & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \quad \text{(carry)} \\ & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ + & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ \hline & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{array}$$

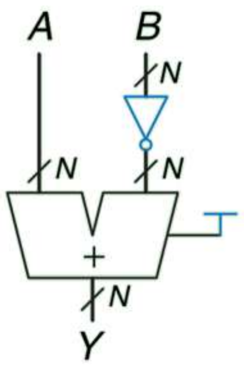
$(10_{10} = 1010_2)$

Subtractor

- Symbol



- Implementation

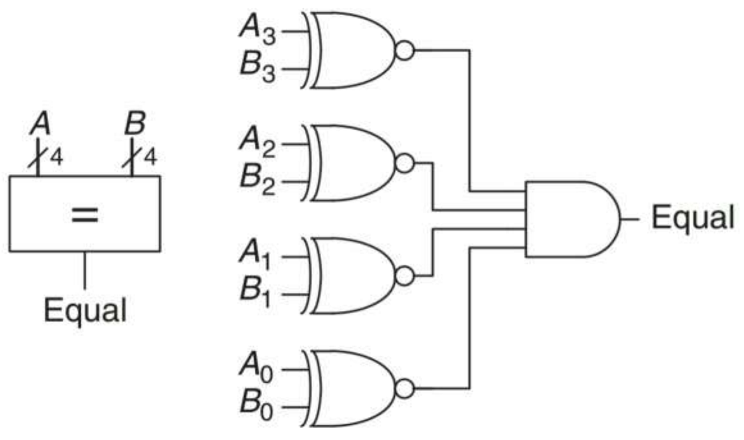


$$Y = A - B = A + \bar{B} + 1$$

Outlines

- Decoder
- **Arithmetic circuits**
  - » Adder
  - » Subtractor
  - » **Comparator**
  - » Arithmetic Logic Unit (ALU)
  - » Multiplier
- How computer works according to the codes?
- Memory arrays
- Logic arrays

Equality Comparator



信息科学与技术学院
School of Information Science and Technology

Magnitude Comparator

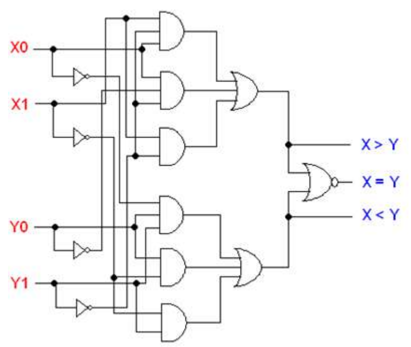
- Truth table of 2-bit magnitude comparator

x_1	x_0	y_1	y_0	x<y	x=y	x>y
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

$$S_{X>Y} = X_1 X_0 \bar{Y}_1 + X_0 \bar{Y}_0 \bar{Y}_1 + X_1 \bar{Y}_1$$

$$S_{X<Y} = \bar{X}_0 Y_0 Y_1 + \bar{X}_1 Y_0 + \bar{X}_1 Y_1$$

$$S_{X=Y} = \overline{S_{X>Y} + S_{X<Y}}$$



信息科学与技术学院
School of Information Science and Technology

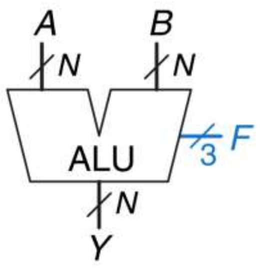
Outlines

- Decoder
- **Arithmetic circuits**
  - » Adder
  - » Subtractor
  - » Comparator
  - » **Arithmetic Logic Unit (ALU)**
  - » Multiplier
- How computer works according to the codes?
- Memory arrays
- Logic arrays

信息科学与技术学院
School of Information Science and Technology

Arithmetic Logic Unit (ALU)

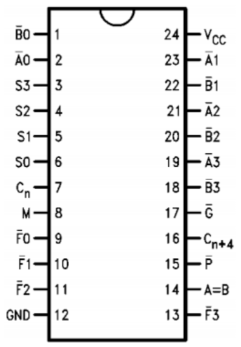
- Combines a variety of mathematical and logical operations into a single unit
- a combinational logic circuit



$F_{2:0}$	Function
000	A AND B
001	A OR B
010	A + B
011	not used
100	A AND $\overline{B}$
101	A OR $\overline{B}$
110	A – B
111	SLT

信息科学与技术学院
School of Information Science and Technology

Example: the 74181, a four-bit ALU

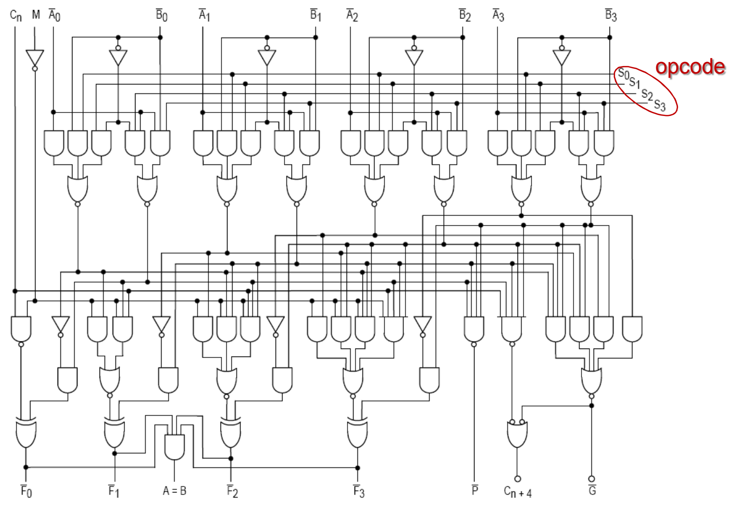


Pin Names	Description
$\overline{A0}$–$\overline{A3}$	Operand Inputs (Active LOW)
$\overline{B0}$–$\overline{B3}$	Operand Inputs (Active LOW)
S0–S3	Function Select Inputs
M	Mode Control Input
C_n	Carry Input
$\overline{F0}$–$\overline{F3}$	Function Outputs (Active LOW)
A = B	Comparator Output
$\overline{G}$	Carry Generate Output (Active LOW)
$\overline{P}$	Carry Propagate Output (Active LOW)
C_{n+4}	Carry Output

Function Table

S3	S2	S1	S0	Logic (M = H)	Arithmetic (Note 2) (M = L) (C_n = L)
L	L	L	L	$\overline{A}$	A minus 1
L	L	L	H	$\overline{AB}$	AB minus 1
L	L	H	L	$\overline{A}+\overline{B}$	$A\overline{B}$ minus 1
L	L	H	H	Logic 1	minus 1
L	H	L	L	$\overline{A}+\overline{B}$	A plus $(A+\overline{B})$
L	H	L	H	$\overline{B}$	AB plus $(A+\overline{B})$
L	H	H	L	$\overline{A}\oplus\overline{B}$	A minus B minus 1
L	H	H	H	$A+\overline{B}$	$A+\overline{B}$
H	L	L	L	$\overline{A}B$	A plus $(A+B)$
H	L	L	H	$A\oplus B$	A plus B
H	L	H	L	B	$A\overline{B}$ plus $(A+B)$
H	L	H	H	$A+B$	$A+B$
H	H	L	L	Logic 0	A plus A (Note 1)
H	H	L	H	$A\overline{B}$	AB plus A
H	H	H	L	AB	$A\overline{B}$ minus A
H	H	H	H	A	A

Realization



信息科学与技术学院
School of Information Science and Technology

Outlines

- Decoder
- Arithmetic circuits
  - » Adder
  - » Subtractor
  - » Comparator
  - » Arithmetic Logic Unit (ALU)
  - » **Multiplier**
- How computer works according to the codes?
- Memory arrays
- Logic arrays

信息科学与技术学院
School of Information Science and Technology

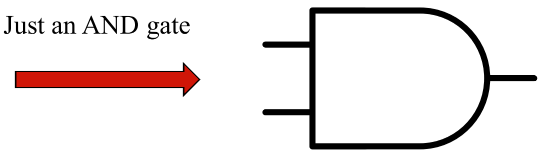
One Bit Multiplication

- Truth table

$$P = A \times B$$

A	B	P
0	0	0
0	1	0
1	0	0
1	1	1

Just an AND gate



信息科学与技术学院
School of Information Science and Technology

More Bits Multiplication

- Decimal
- Binary

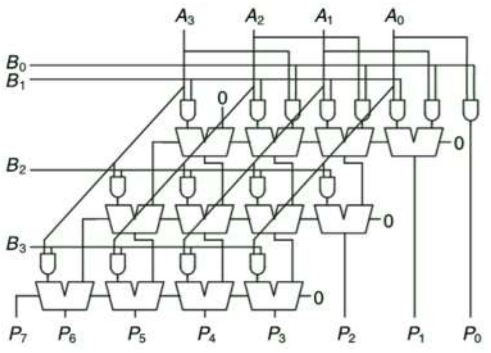
$$\begin{array}{rrl}
 & 230 & \text{multiplicand} \\
\times & 42 & \text{multiplier} \\
\hline
 & 460 & \text{partial} \\
+ & 920 & \text{products} \\
\hline
 & 9660 & \text{result}
\end{array}
\qquad
\begin{array}{rr}
 & 0101 \\
\times & 0111 \\
\hline
 & 0101 \\
 & 0101 \\
 & 0101 \\
+ & 0000 \\
\hline
 & 0100011
\end{array}$$

$2\ 30 \times 42 = 9660$ $\qquad$ $5 \times 7 = 35$

信息科学与技术学院
School of Information Science and Technology

Implementation

$$\begin{array}{rcccccccc}
 & & & & & A_3 & A_2 & A_1 & A_0 \\
\times & & & & & B_3 & B_2 & B_1 & B_0 \\
\hline
 & & & & & A_3B_0 & A_2B_0 & A_1B_0 & A_0B_0 \\
 & & & & A_3B_1 & A_2B_1 & A_1B_1 & A_0B_1 & \\
 & & & A_3B_2 & A_2B_2 & A_1B_2 & A_0B_2 & & \\
+ & & A_3B_3 & A_2B_3 & A_1B_3 & A_0B_3 & & & \\
\hline
 & P_7 & P_6 & P_5 & P_4 & P_3 & P_2 & P_1 & P_0
\end{array}$$



信息科学与技术学院
School of Information Science and Technology

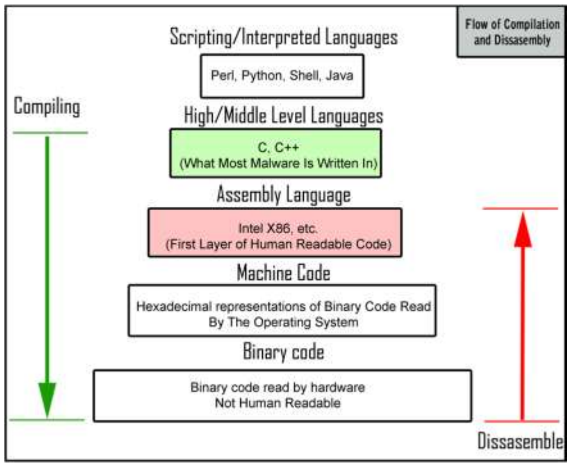
Outlines

- Decoder
- Arithmetic circuits
  - » Adder
  - » Subtractor
  - » Comparator
  - » Arithmetic Logic Unit (ALU)
  - » Multiplier
- **How computer works according to the codes?**
- Memory arrays
- Logic arrays

信息科学与技术学院
School of Information Science and Technology

Assembly Language (汇编语言)

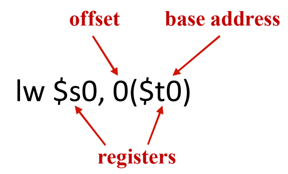
- A low-level programming language
- First layer of human readable code
- Strong correspondence to particular computer architecture's machine code instructions
- Example: *MIPS*, a reduced instruction set computer (RISC) instruction set architecture (ISA)


Flow of Compilation and Dissasembly
Scripting/Interpreted Languages
Perl, Python, Shell, Java
Compiling
High/Middle Level Languages
C, C++
(What Most Malware Is Written In)
Assembly Language
Intel X86, etc.
(First Layer of Human Readable Code)
Machine Code
Hexadecimal representations of Binary Code Read By The Operating System
Binary code
Binary code read by hardware
Not Human Readable
Dissasemble

Example: Doing Addition with MIPS

- In C or Java

  z = **w** + **y**;



- With MIPS

```
la $t0, w              # put address of w into $t0
lw $s0, 0($t0)         # put contents of w into $s0
la $t1, y              # put address of y into $t1
lw $s1, 0($t1)         # put contents of y into $s1
add $s2, $s0, $s1      # add w + y, put result in $s2
la $t2, z              # put address of z into $t2
sw $s2, 0($t2)         # put contents of $s2 into z
```

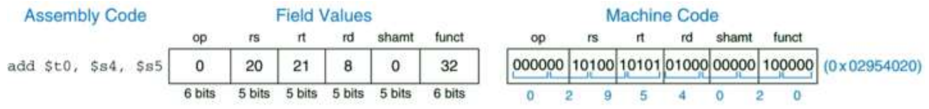
MIPS Instruction

- Instruction set

Type	31 ….			format (bits)		…. 0
R	opcode (6)	rs (5)	rt (5)	rd (5)	shamt (5)	funct (6)
I	opcode (6)	rs (5)	rt (5)	immediate (16)		
J	opcode (6)	address (26)				

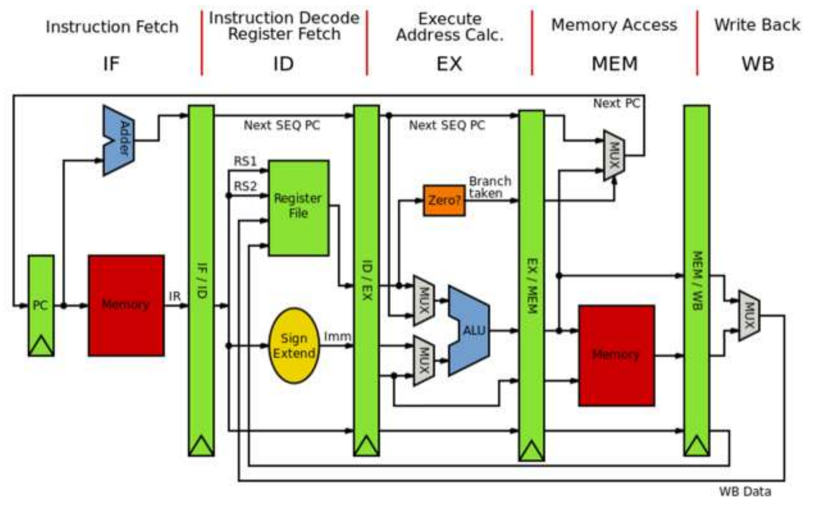
  » R for registers; I for immediate value; J for jump

- Examples



信息科学与技术学院 School of Information Science and Technology

MIPS Pipeline



信息科学与技术学院 School of Information Science and Technology

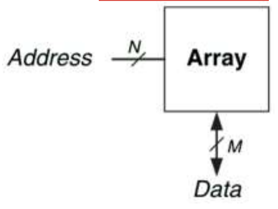
Outlines

- Decoder
- Arithmetic circuits
  - » Adder
  - » Subtractor
  - » Comparator
  - » Arithmetic Logic Unit (ALU)
  - » Multiplier
- How computer works according to the codes?
- **Memory arrays**
- Logic arrays

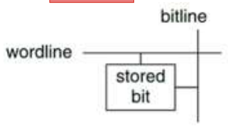
Memory array

- Two-dimensional array of memory cells
- The row is specified as *Address*
- Each row of data is called a *Word*
- The array contains 2^N *M-bit* words
- *Depth* = 2^N
- *Width* = 2^M
- Types:
  - » Dynamic random access memory (DRAM)
  - » Static random access memory (SRAM)
  - » Read only memory (ROM)
  - » etc.

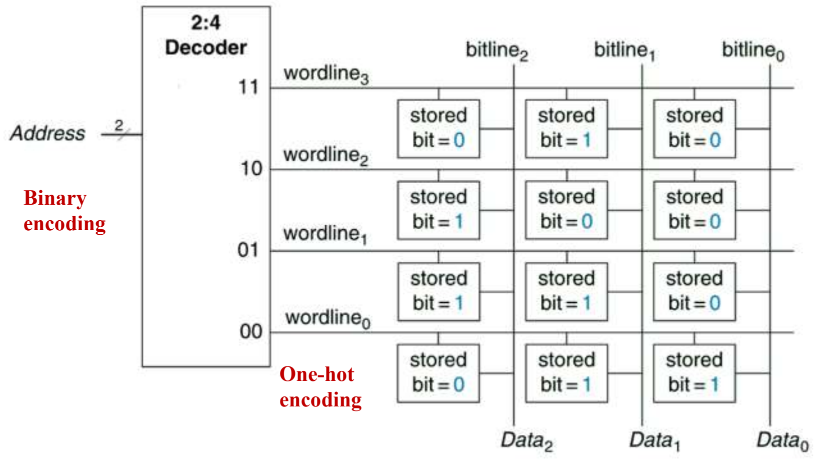
General symbol



Bit cell



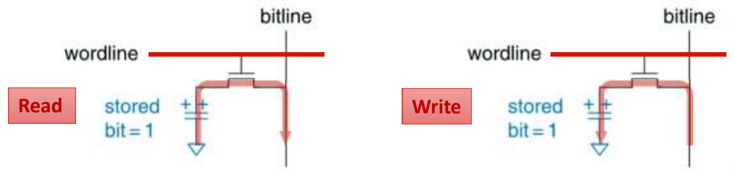
Example: 4 × 3 Memory Array



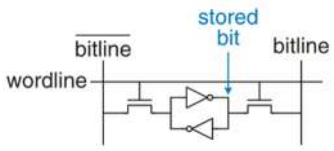
信息科学与技术学院
School of Information Science and Technology

Volatile Memory

- Dynamic Random Access Memory (DRAM)



- Static Random Access Memory (SRAM)



信息科学与技术学院
School of Information Science and Technology

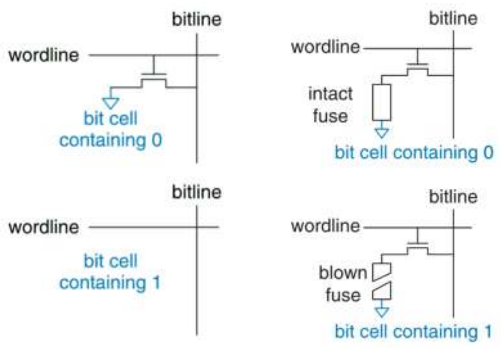
Memory Comparison

- The best memory type for a particular design depends on the *speed*, *cost*, and *power constraints*.

Memory Type	Transistors per Bit Cell	Latency
flip-flop	~20	fast
SRAM	6	medium
DRAM	1	slow

Nonvolatile Memory

- Read only memory (ROM)



ROM bit cells **Programmable ROM**

- modern ROMs can programmed (written) as well.
- Generally, ROMs take a longer time to write then RAMs, but are nonvolatile.

Outlines

- Decoder
- Arithmetic circuits
  - » Adder
  - » Subtractor
  - » Comparator
  - » Arithmetic Logic Unit (ALU)
  - » Multiplier
- How computer works according to the codes?
- Memory arrays
- **Logic arrays**

信息科学与技术学院
School of Information Science and Technology

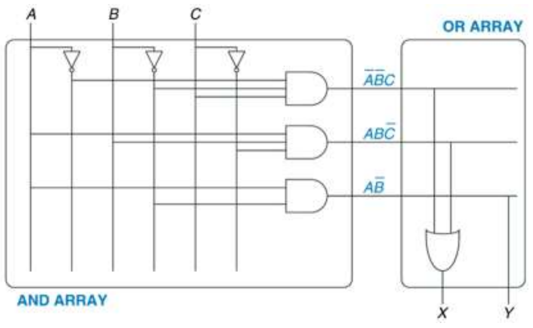
Logic array

- Gates organized into regular arrays, whose connections are programmable
  - » Reconfigurable by software tools
  - » Mass produced in large quantities
  - » Inexpensive
  - » Fast I/Os
- Technologies
  - » Programmable logic arrays (PLAs)
    - – Older technologies
    - – perform only combinational logic functions
  - » Field programmable gate arrays (FPGAs)
    - – perform both combinational and sequential logic

信息科学与技术学院
School of Information Science and Technology

Programmable logic arrays (PLAs)

- Two-level combinational logic in sum-of-products (SOP) form
- built from an AND array followed by an OR array

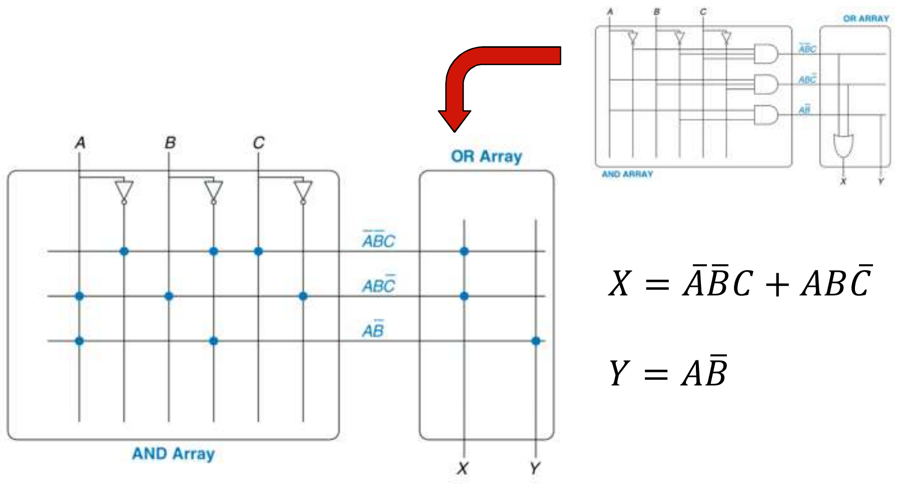


$$X = \bar{A}\bar{B}C + AB\bar{C}$$

$$Y = A\bar{B}$$

信息科学与技术学院
School of Information Science and Technology

Dot Notation



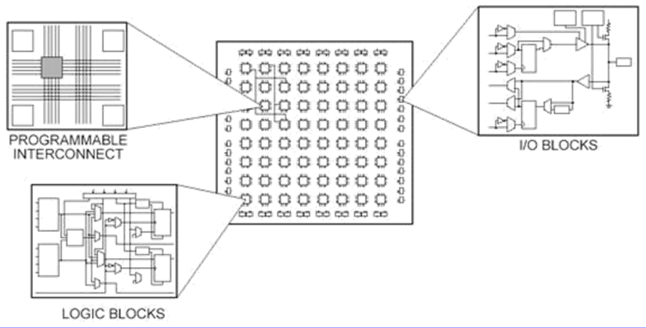
$$X = \bar{A}\bar{B}C + AB\bar{C}$$

$$Y = A\bar{B}$$

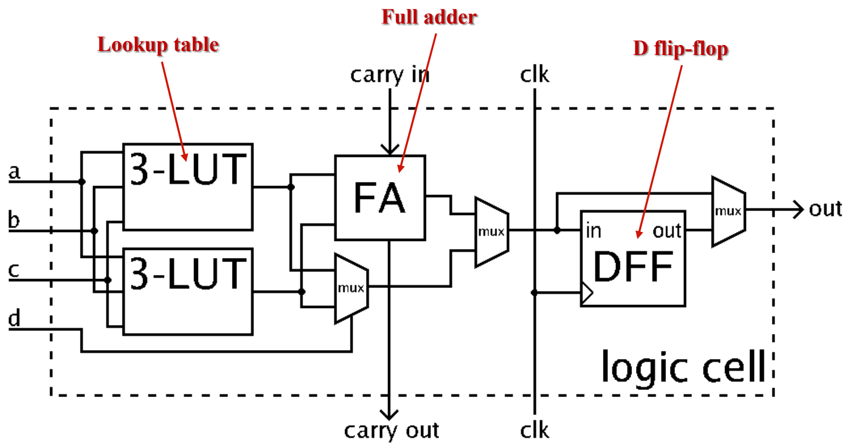
信息科学与技术学院
School of Information Science and Technology

Field programmable gate arrays (FPGAs)

- More powerful and more flexible than PLAs
- Can implement both combinational and sequential logic
- Can also implement multilevel logic functions
- Integrate other useful features such as built-in multipliers, high-speed I/Os, large RAM arrays, processors, etc.



A Logic Cell


Full adder
Lookup table
D flip-flop
carry in
clk
a
b
c
d
3-LUT
3-LUT
FA
mux
mux
mux
in
out
DFF
out
logic cell
carry out
clk