Digital Electronics and Microprocessors

Class 20

CHHAYADEVI BHAMARE

Digital Arithmetic: Operations and Circuits(Chapter 6)

Arithmetic operations

□ Binary addition

□ Representing Signed Numbers

□ Addition and subtractionin the 2's complement systems.

□ Multiplication and division of binary numbers

□ BCD addition

□ Hexadecimal Arithmetic

Digital Arithmetic: Operations and Circuits(Chapter 6)

Arithmetic Circuits

□ Design of a full adder

□ Parallel binary adder

□ Complete parallel adder with registers

□ Carry propogation

□ IC arithmetic circuits (parallele adder, cascading of parallel adder, ALU IC etc)

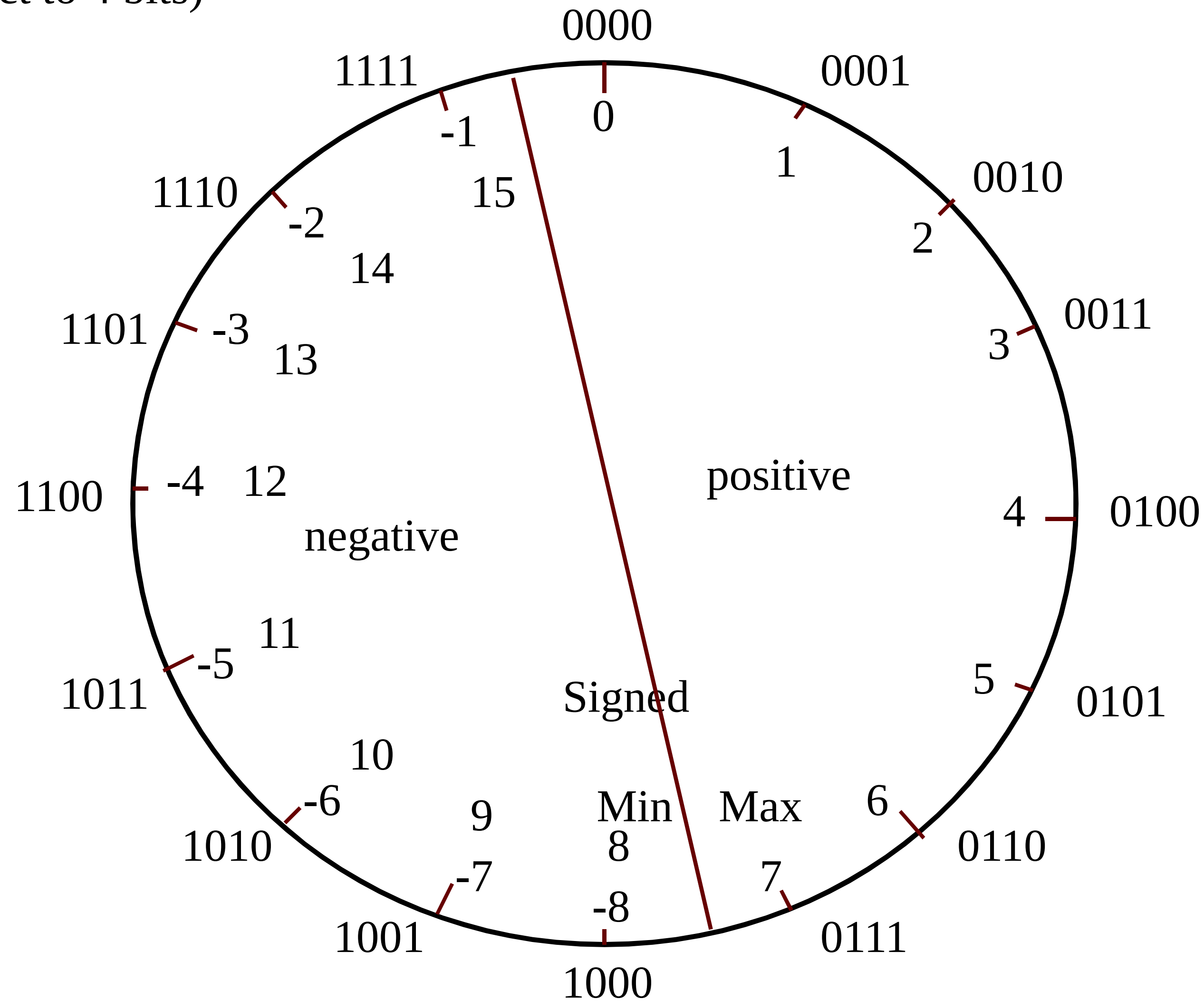
Signed binary numbers: Possible representations

□ Sign Magnitude:	One's Complement	Two's Complement
000 = +0	000 = +0	000 = +0
001 = +1	001 = +1	001 = +1
010 = +2	010 = +2	010 = +2
011 = +3	011 = +3	011 = +3
100 = -0	100 = -3	100 = -4
101 = -1	101 = -2	101 = -3
110 = -2	110 = -1	110 = -2
111 = -3	111 = -0	111 = -1

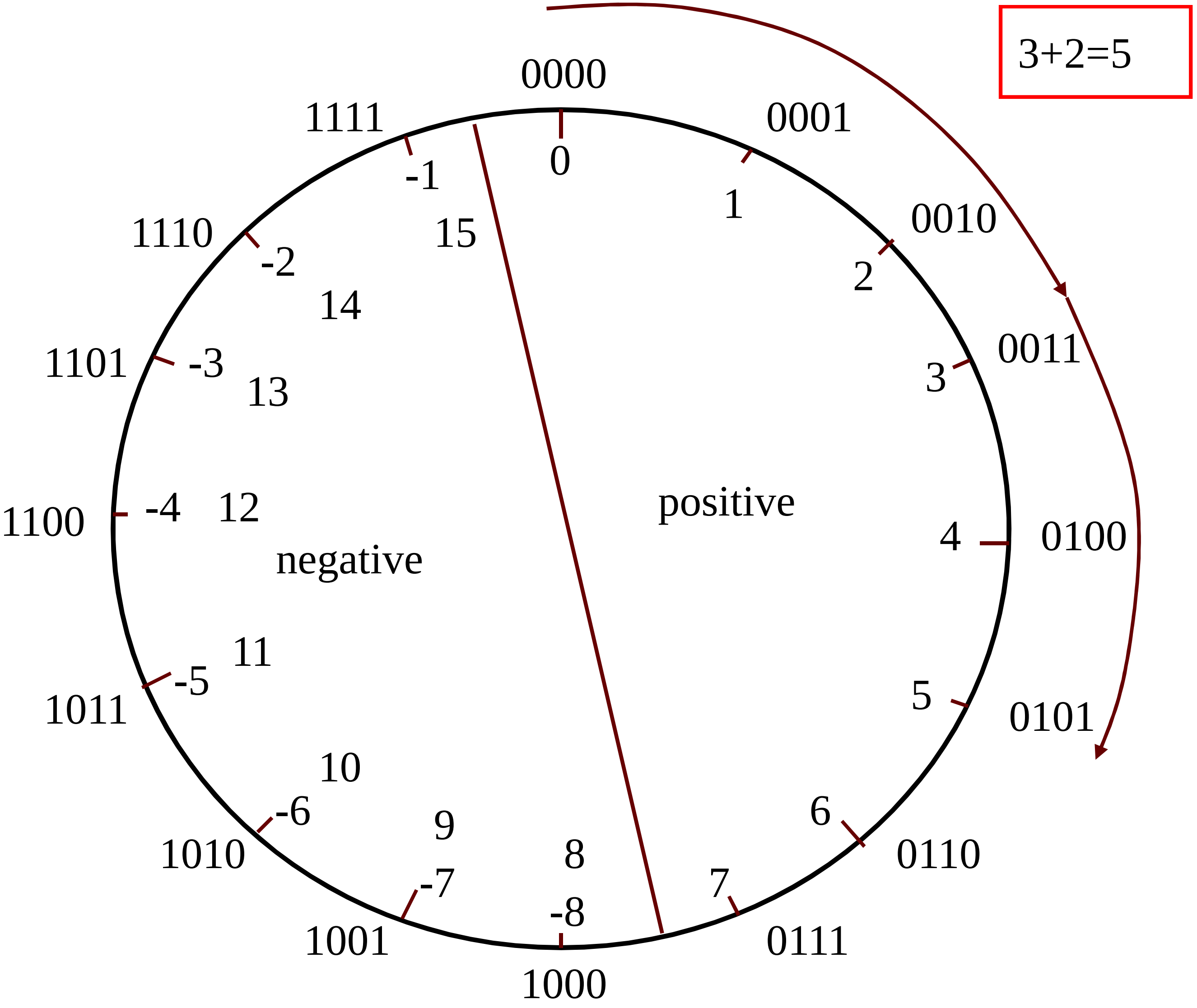
- □ Issues: balance, number of zeros, ease of operations
- □ Which one is best? Why?
- □ Pick the representation that made the hardware simple

Two’s complement

(let’s restrict to 4 bits)



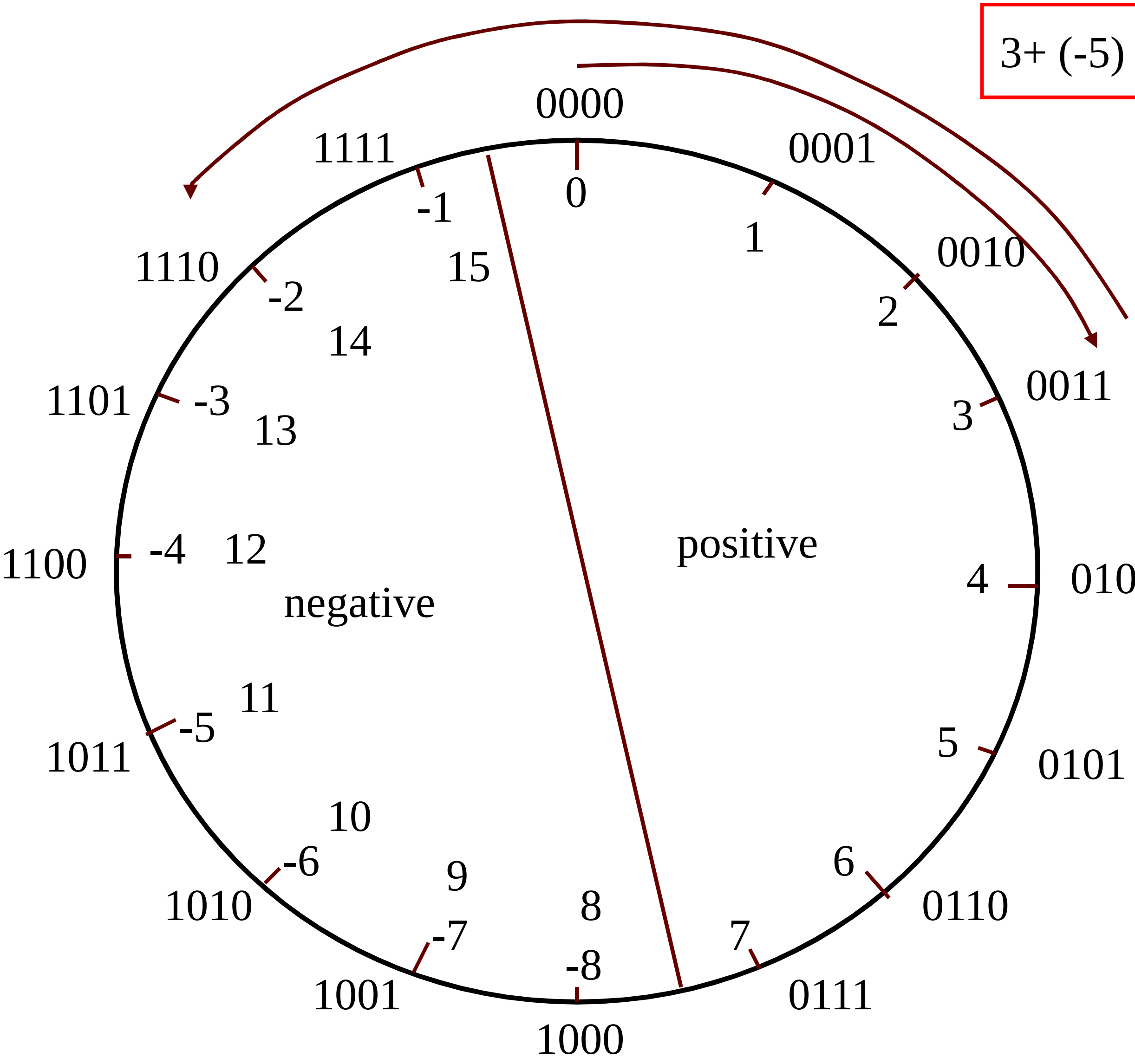
Two’s complement



3+2=5

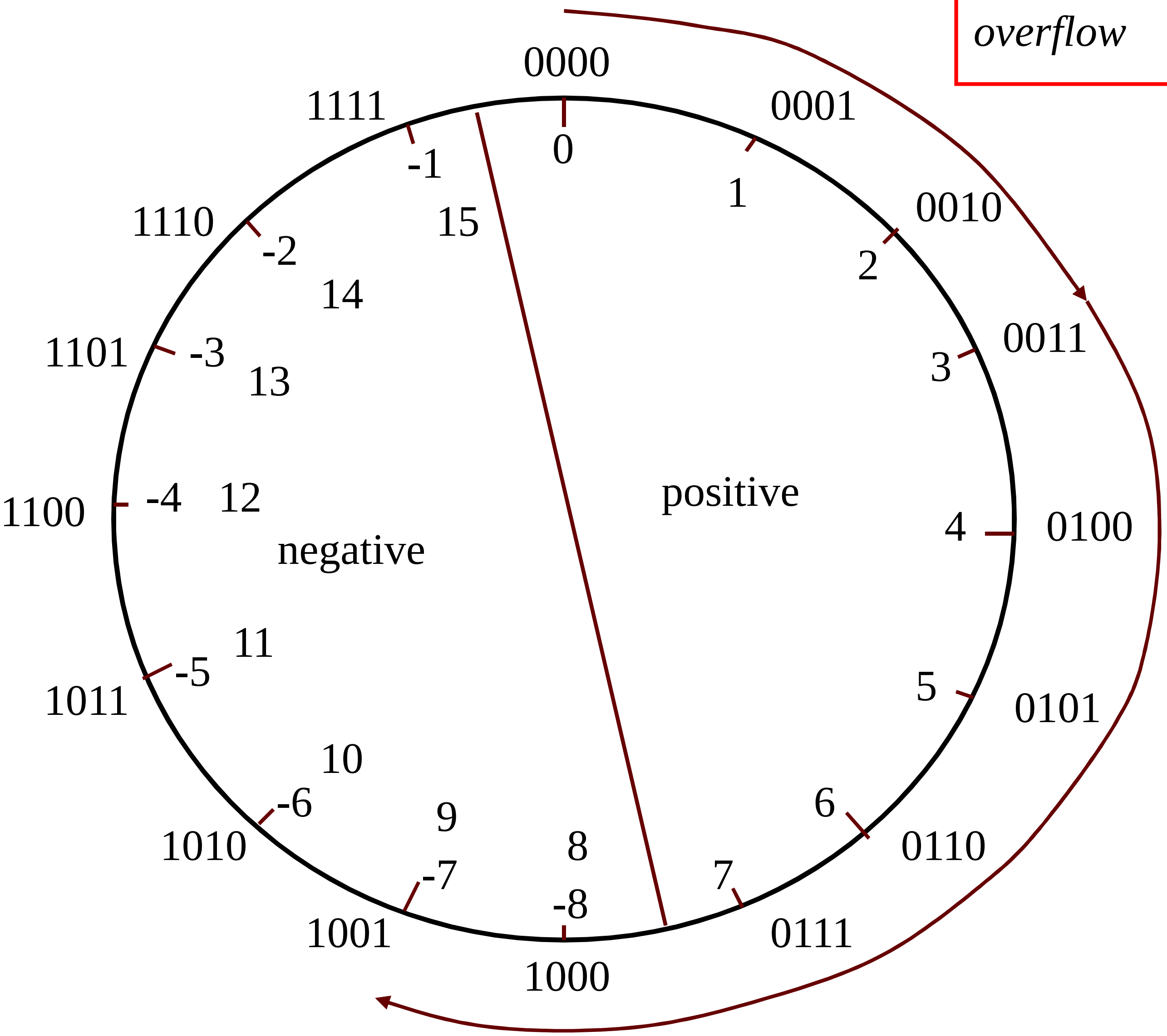
Two’s complement

3+ (-5) = -2



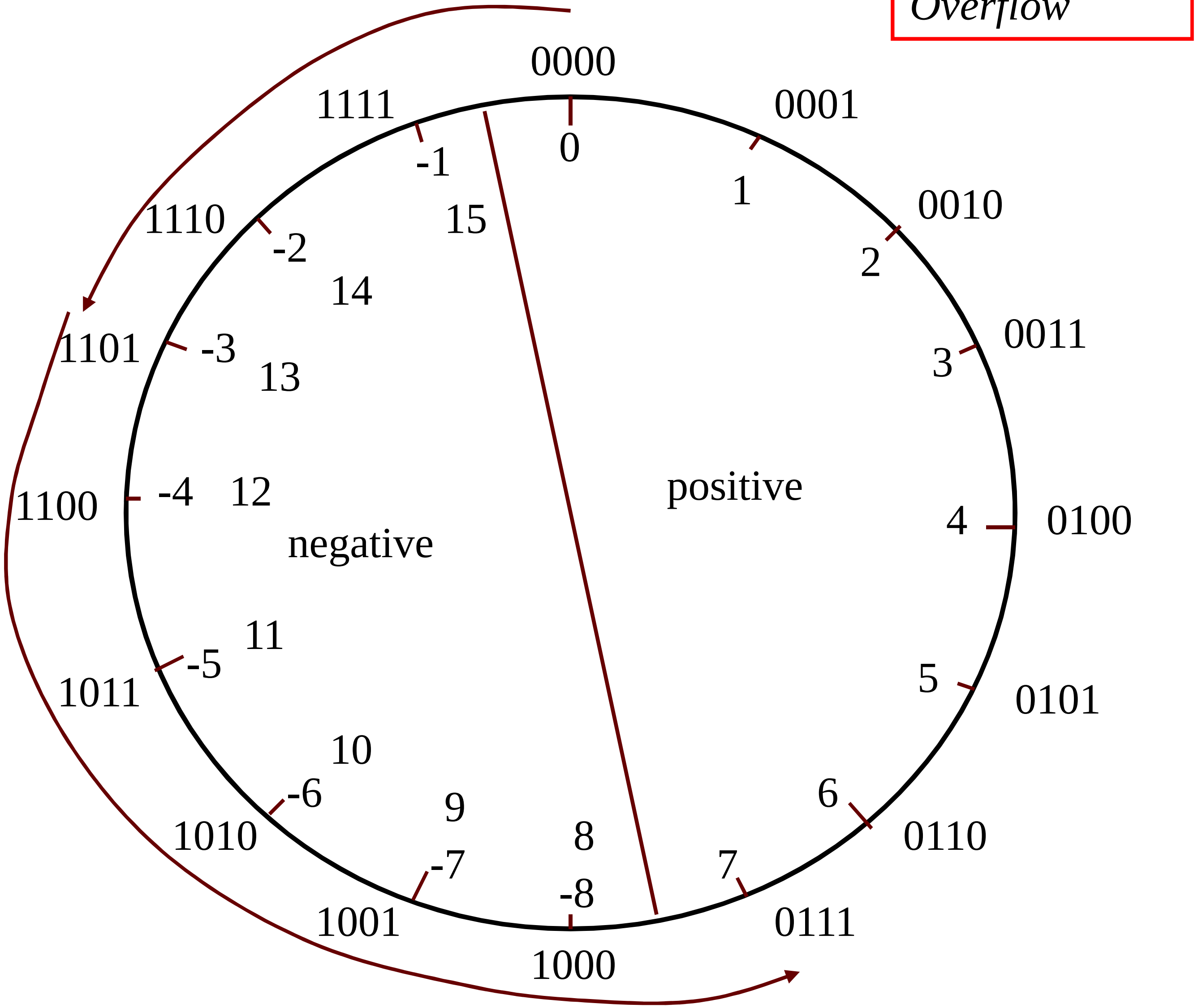
Two's complement

3+6 = -7 !!
overflow

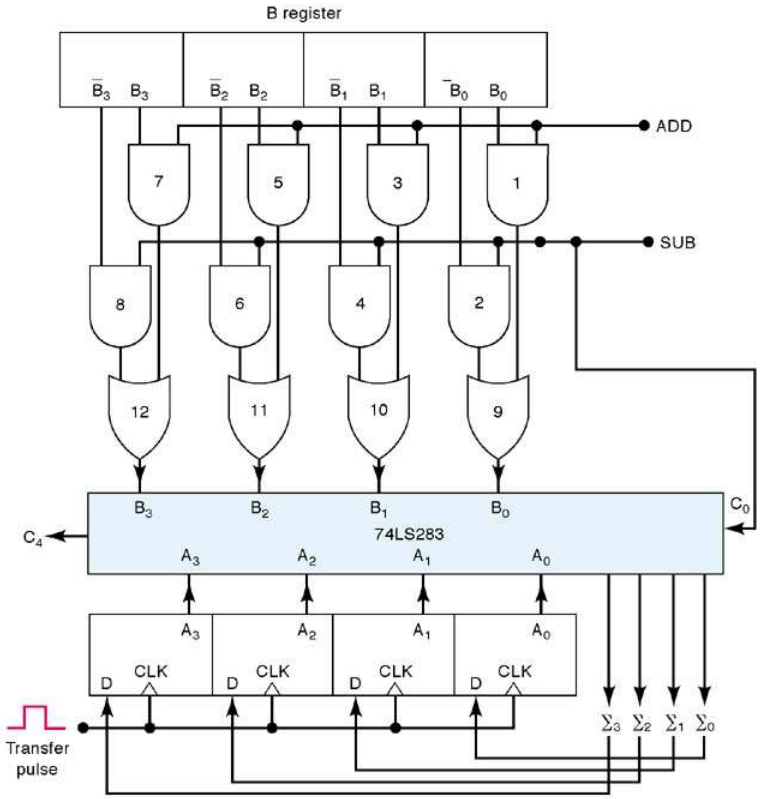


Two’s complement

-3 + (-6) = 7 !!
Overflow



Combined Addition and Subtraction



BCD ADDER

- Add the BCD code groups for each decimal digit position; use ordinary binary addition.
- For those positions where the sum is 9 or less, the sum is in proper BCD form and no correction is needed
- When the sum of two digits is greater than 9, a correction of 0110 should be added to that sum to produce the proper BCD result. This will produce a carry to be added to the next decimal position.
  - $A_3A_2A_1A_0$ ← BCD code group
  - $\underline{B_3B_2B_1B_0}$ ← BCD code group

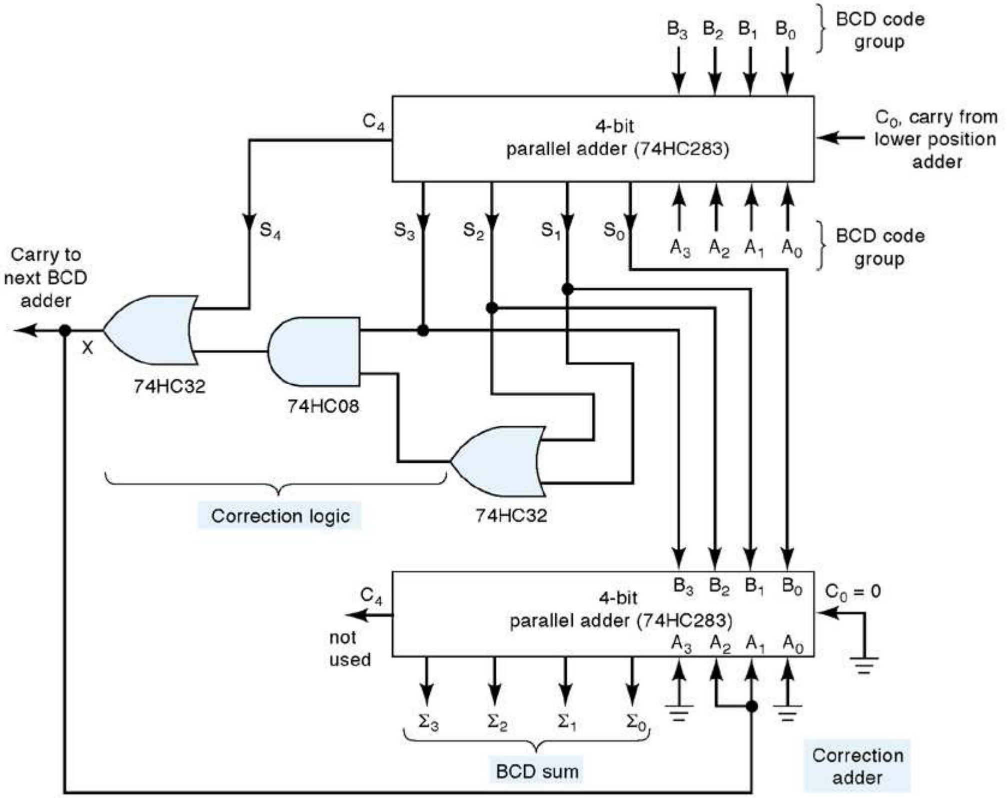
  $S_4S_3S_2S_1S_0$ ← straight binary sum

Truth table

S_4	S_3	S_2	S_1	S_0	
0	1	0	1	0	(10)
0	1	0	1	1	(11)
0	1	1	0	0	(12)
0	1	1	0	1	(13)
0	1	1	1	0	(14)
0	1	1	1	1	(15)
1	0	0	0	0	(16)
1	0	0	0	1	(17)
1	0	0	1	0	(18)

$X=S_4+S_3(S_2+S_1)$

A BCD adder



B_3 B_2 B_1 B_0
BCD code group
C_4
4-bit parallel adder (74HC283)
C_0, carry from lower position adder
S_4
S_3
S_2
S_1
S_0
A_3 A_2 A_1 A_0
BCD code group
Carry to next BCD adder
X
74HC32
74HC08
74HC32
Correction logic
B_3 B_2 B_1 B_0
$C_0 = 0$
C_4
not used
4-bit parallel adder (74HC283)
A_3 A_2 A_1 A_0
Σ_3 Σ_2 Σ_1 Σ_0
BCD sum
Correction adder

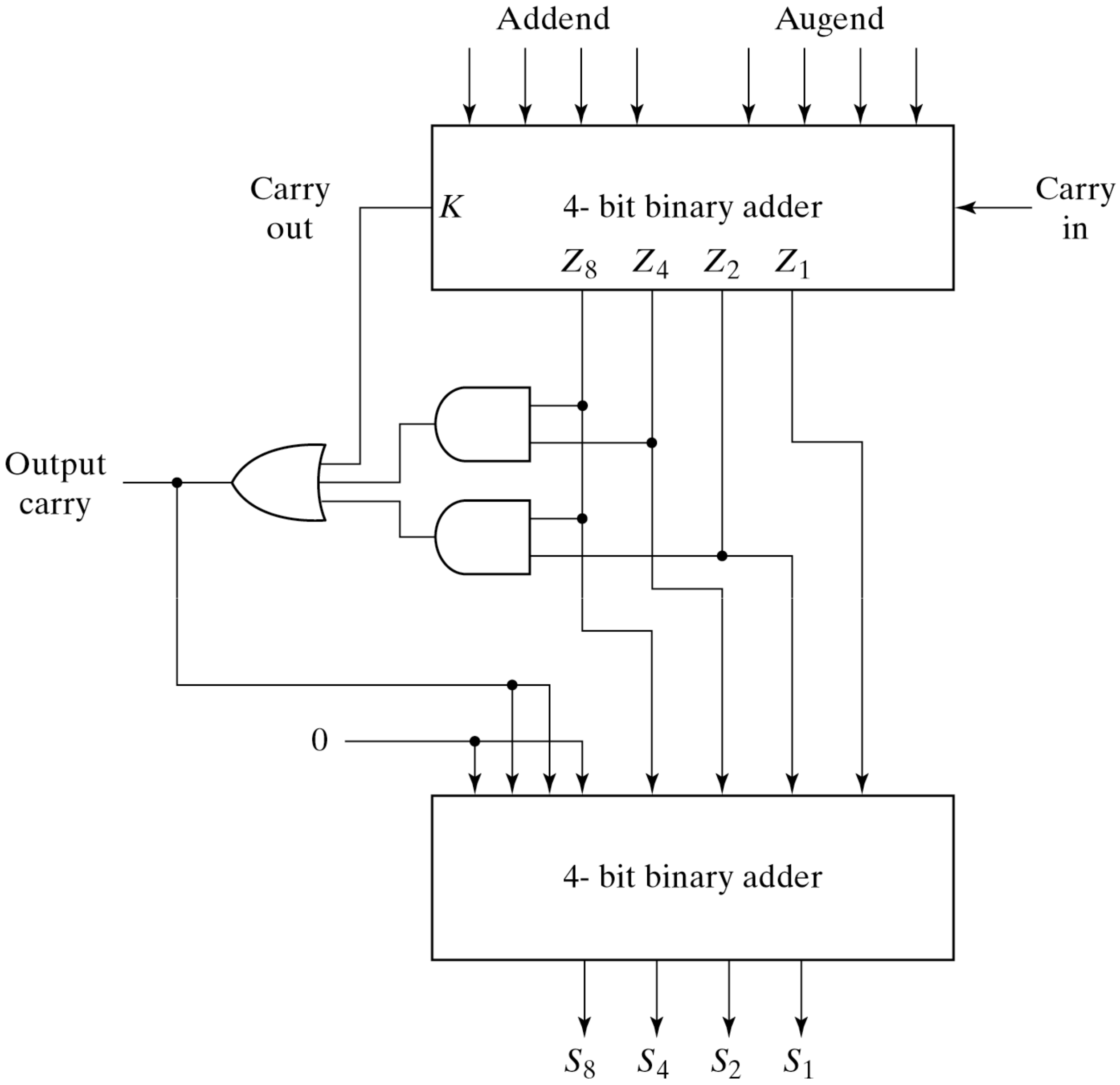
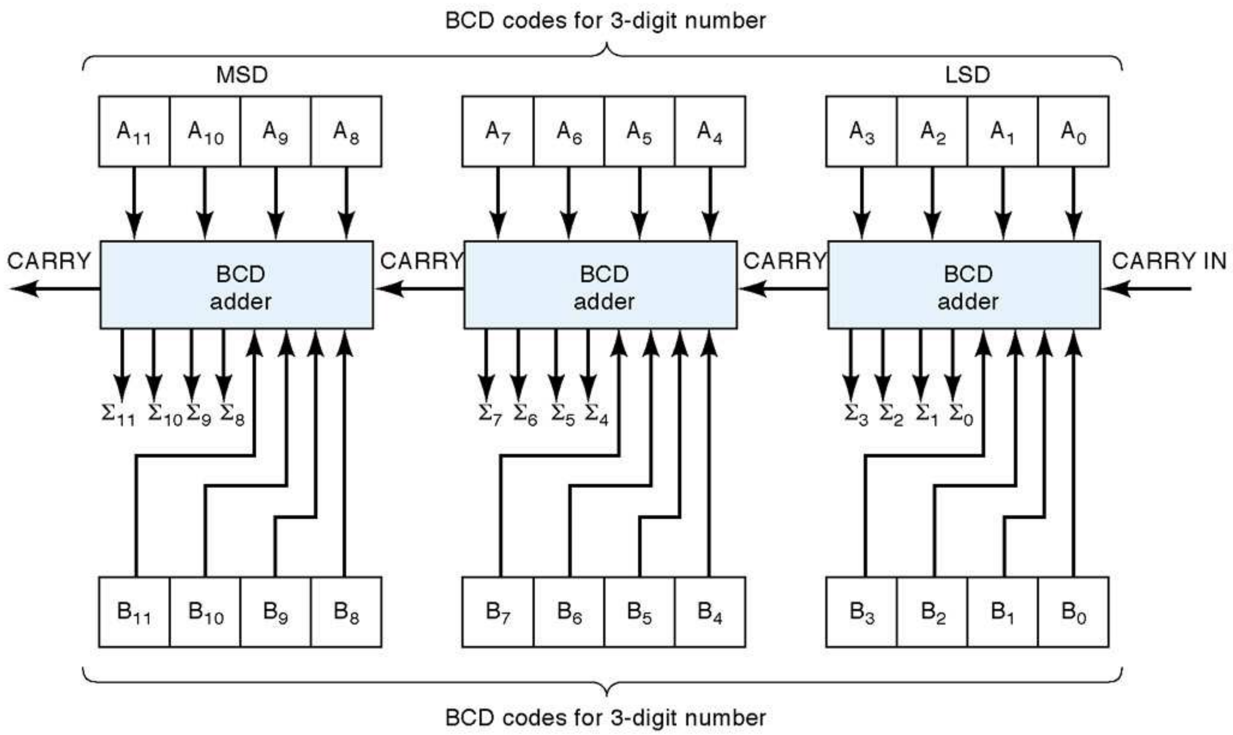


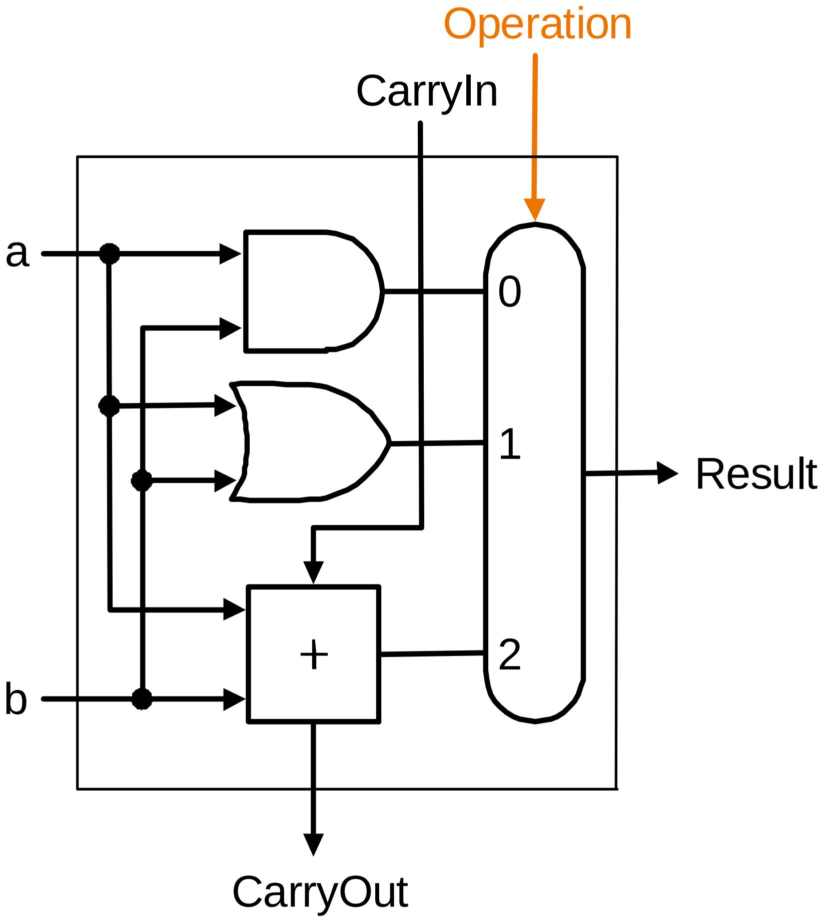
Fig. 4-14 Block Diagram of a BCD Adder

Example

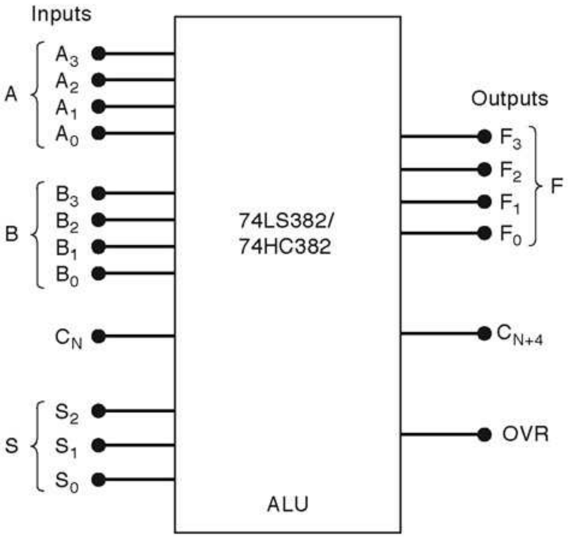
- Determine the inputs and outputs when the below circuit is used to add 538_{10} to 247_{10}. Assume CARRY IN=0.


BCD codes for 3-digit number
MSD
LSD
A_{11} A_{10} A_9 A_8
A_7 A_6 A_5 A_4
A_3 A_2 A_1 A_0
CARRY
BCD adder
CARRY
BCD adder
CARRY
BCD adder
CARRY IN
Σ_{11} Σ_{10} Σ_9 Σ_8
Σ_7 Σ_6 Σ_5 Σ_4
Σ_3 Σ_2 Σ_1 Σ_0
B_{11} B_{10} B_9 B_8
B_7 B_6 B_5 B_4
B_3 B_2 B_1 B_0
BCD codes for 3-digit number

ALU(1 bit)



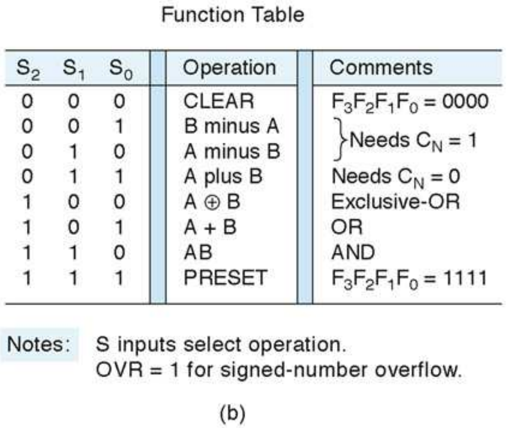
ALU(4-Bit) Integrated Circuits



A = 4-bit input number
B = 4-bit input number
C_N = carry into LSB position
S = 3-bit operation select inputs

F = 4-bit output number
C_{N+4} = carry out of MSB position
OVR = overflow indicator

(a)



Function Table

S_2	S_1	S_0	Operation	Comments
0	0	0	CLEAR	$F_3F_2F_1F_0 = 0000$
0	0	1	B minus A	Needs $C_N = 1$
0	1	0	A minus B	
0	1	1	A plus B	Needs $C_N = 0$
1	0	0	$A \oplus B$	Exclusive-OR
1	0	1	A + B	OR
1	1	0	AB	AND
1	1	1	PRESET	$F_3F_2F_1F_0 = 1111$

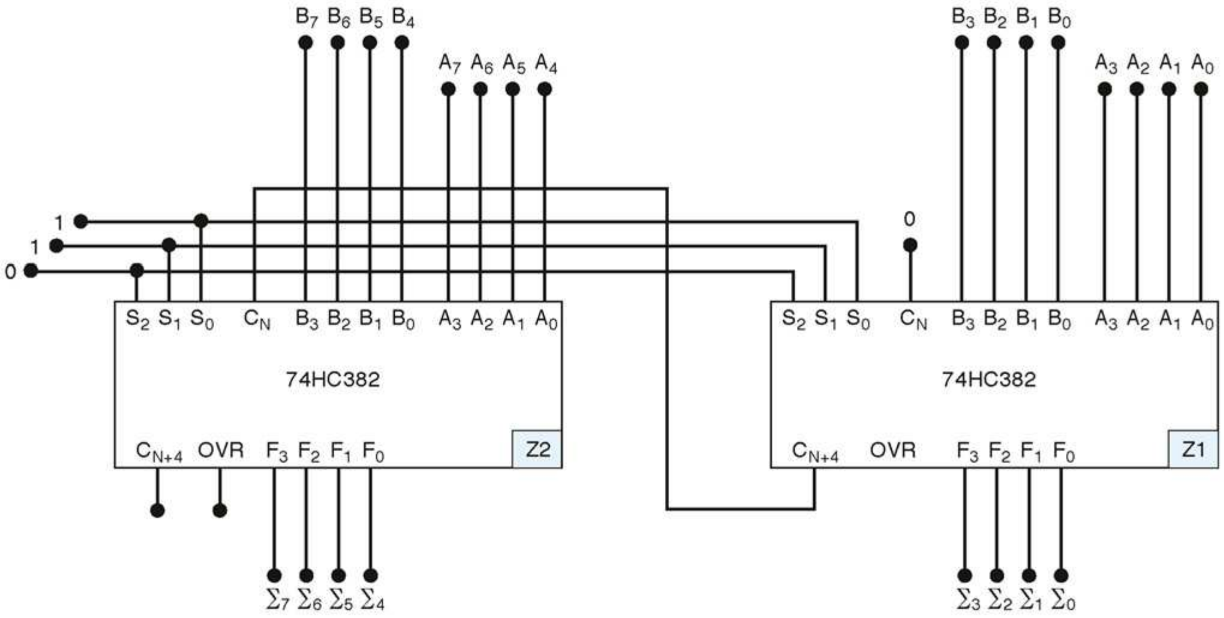
Notes: S inputs select operation.
OVR = 1 for signed-number overflow.

(b)

Operations

- CLEAR
- ADD
- SUBTRACT
- XOR
- OR
- AND
- PRESET

Expanding the ALU



B_7 B_6 B_5 B_4
A_7 A_6 A_5 A_4
B_3 B_2 B_1 B_0
A_3 A_2 A_1 A_0
1
1
0
0
S_2 S_1 S_0 C_N B_3 B_2 B_1 B_0 A_3 A_2 A_1 A_0
74HC382
C_{N+4} OVR F_3 F_2 F_1 F_0
Z2
S_2 S_1 S_0 C_N B_3 B_2 B_1 B_0 A_3 A_2 A_1 A_0
74HC382
C_{N+4} OVR F_3 F_2 F_1 F_0
Z1
Σ_7 Σ_6 Σ_5 Σ_4
Σ_3 Σ_2 Σ_1 Σ_0

Notes: Z1 adds lower-order bits.
Z2 adds higher-order bits.
Σ_7–Σ_0 = 8-bit sum.
OVR of Z2 is 8-bit overflow indicator.

Assignment 3

- 6-26 to 6-39