



CSCI 150

Introduction to Digital and Computer System Design

Lecture 3: Combinational Logic Design VIII



Jetic Gū

Overview

- Focus: Arithmetic Functional Blocks
- Architecture: Combinatory Logical Circuits
- Textbook v4: Ch4 4.4, 4.5; v5: Ch2 2.9, Ch3 3.11, 3.12
- Core Ideas:
 1. Overflow
 2. Signed Arithmetics
 3. Other Functions

Adder Subtractor Units (Unsigned)

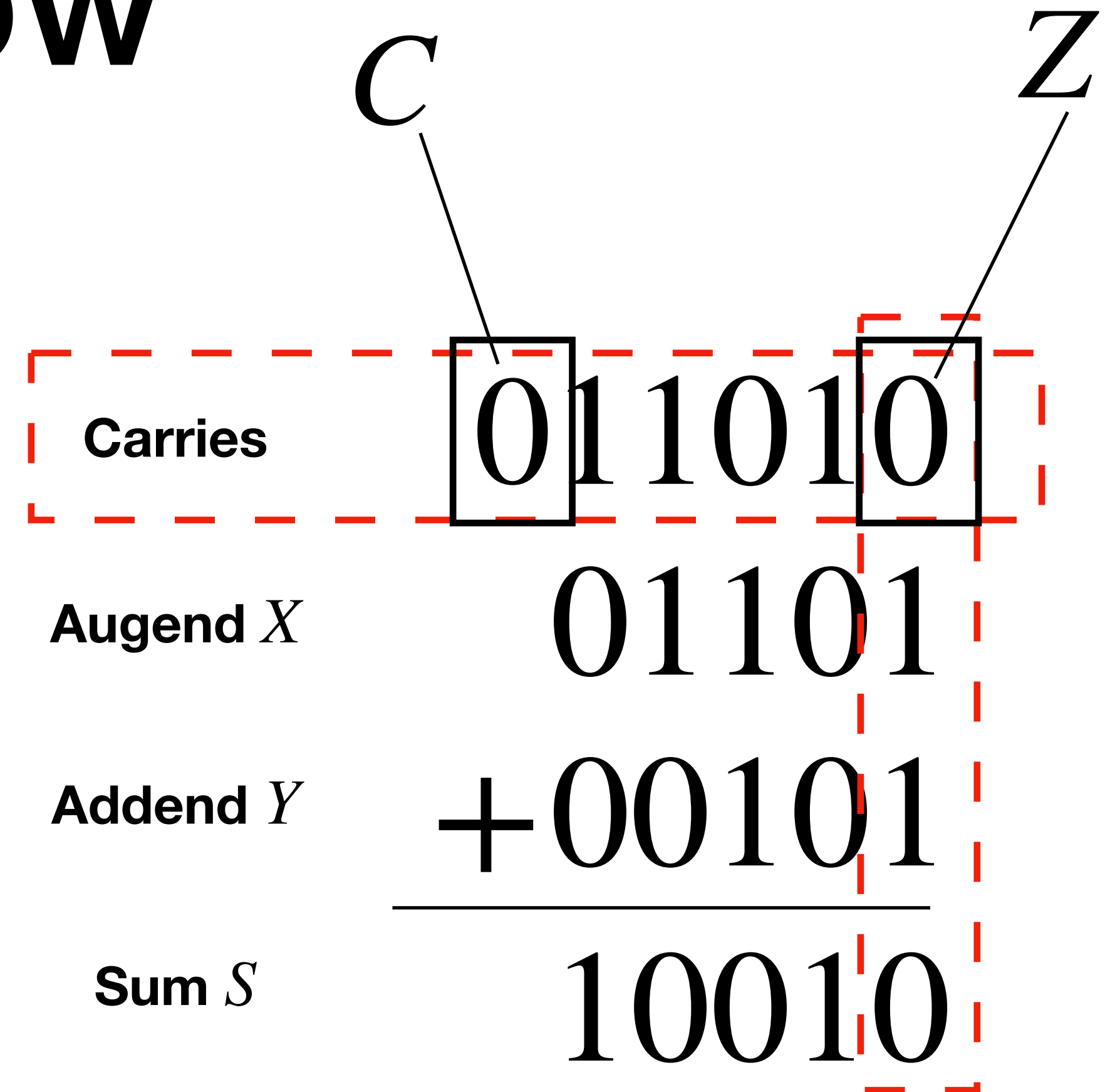
- Binary Adder: 1-bit Half Adder; 1-bit Full Adder; n -bit Adder
- Binary Subtractor: 1-bit Subtractor; n -bit subtractor
- 2s complement
- Binary Adder-Subtractor Unit
using Adder, Subtractor, Complementer and Multiplexer
- Binary Adder-Subtractor Unit
using Adder and XOR

Overflow

- If we start with 2 n -bit numbers, but the result requires more than n -bits, then there is an overflow

Overflow

- This is 5 5-bit addition
- If the carry bit $C = 0$, there is no overflow
- If the carry bit $C = 1$, there is overflow!



Overflow

- In 4-bit unsigned binary addition, does $7 + 8$ cause overflow?
- Does $10 + 7$?

Overflow

- In 8-bit unsigned binary Multiplication, does 12×12 cause overflow?
- Does 17×17 ?

Signed Arithmetics Functions

Difference

- Unsigned n -bit Integer
 - n -bits for actual value (magnitude): $[0, 2^n - 1]$
- Signed n -bit Integer
 - Leftmost bit for sign: 0 for positive and zero; 1 for negative
 - $n - 1$ -bits for actual value: $[0, 2^{n-1} - 1]$ (sign=0), $[-2^{n-1}, -1]$ (sign=1)

Signed 2s Complement

- 2s Complement of X ($X \geq 0$)
 - $2^n - X$
- Signed 2s Complement of X
 - Sign=0, X
 - Sign=1, $2^{n-1} + X$
Invert magnitude bits then plus 1

!!!! 2s complement and Signed 2s complement are different!

Decimal	Signed	Signed 2s Complement
7	0111	0111
6	0110	0110
5	0101	0101
4	0100	0100
3	0011	0011
2	0010	0010
1	0001	0001
0	0000	0000
-1	1001	1111
-2	1010	1110
-3	1011	1101
-4	1100	1100
-5	1101	1011
-6	1110	1010
-7	1111	1001
-8	1000	1000

Concept

Signed 2s Complement

- Let X be a positive number
- Signed 2s complement of X is X
- Signed 2s complement of $-X$ is the 2s complement of X
- 2s complement of [Signed 2s complement of $-X$] is $-X$

Decimal	Signed	Signed 2s Complement
7	0111	0111
6	0110	0110
5	0101	0101
4	0100	0100
3	0011	0011
2	0010	0010
1	0001	0001
0	0000	0000
-1	1001	1111
-2	1010	1110
-3	1011	1101
-4	1100	1100
-5	1101	1011
-6	1110	1010
-7	1111	1001
-8	1000	1000

Identical

Signed 2s Complement Addition

- In Signed 2s complement, addition is like unsigned
- Step1: convert the two numbers to **Signed 2s Complement**
- Step2: perform addition using unsigned adder
- Step3: output the **Signed 2s Complement** of the result from Step2
- $1101 + 0011 = 0000$
(Decimal) $-3 + 3 = 0$

Decimal	Signed	Signed 2s Complement
7	0111	0111
6	0110	0110
5	0101	0101
4	0100	0100
3	0011	0011
2	0010	0010
1	0001	0001
0	0000	0000
-1	1001	1111
-2	1010	1110
-3	1011	1101
-4	1100	1100
-5	1101	1011
-6	1110	1010
-7	1111	1001
-8	1000	1000

Signed 2s Complement Subtraction

- In Signed 2s complement, subtraction is also like unsigned
- Step1: convert the two numbers to **Signed 2s Complement**
- Step2: perform addition using unsigned subtractor
- Step3: output the **Signed 2s Complement** of the result from Step2
- $0000 - 0001 = 1111$
(Decimal) $0 - 1 = -1$

Decimal	Signed	Signed 2s Complement
7	0111	0111
6	0110	0110
5	0101	0101
4	0100	0100
3	0011	0011
2	0010	0010
1	0001	0001
0	0000	0000
-1	1001	1111
-2	1010	1110
-3	1011	1101
-4	1100	1100
-5	1101	1011
-6	1110	1010
-7	1111	1001
-8	1000	1000

Signed 2s Complement Addition

- Perform signed 8-bit binary addition of $12 + 15$ 0000 1100 + 0000 1111
- Perform signed 8-bit binary addition of $-12 + 15$ 1111 0100 + 0000 1111
- Perform signed 8-bit binary addition of $12 + (-15)$ 0000 1100 + 1111 0001
- Perform signed 8-bit binary addition of $-12 + (-15)$ 1111 0100 + 1111 0001

Signed 2s Complement Subtraction

- Perform signed 8-bit binary Subtraction of $17 - 14$ 0001 0001 - 0000 1110
- Perform signed 8-bit binary Subtraction of $-17 - 14$ 1110 1111 - 0000 1110
- Perform signed 8-bit binary Subtraction of $17 - (-14)$ 0001 0001 - 1111 0010
- Perform signed 8-bit binary Subtraction of $-17 - (-14)$ 1110 1111 - 1111 0010

Think about it

- Can signed n -bit binary addition cause overflow?
 - Pos + Pos
 - Pos + Neg (Neg + Pos)
 - Neg + Neg
- Can signed n -bit binary subtraction cause overflow?

Signed 2s Complement Addition

- Perform signed 6-bit binary addition of $10 + 13$
- Perform signed 6-bit binary addition of $-10 + 13$
- Perform signed 6-bit binary addition of $10 + (-13)$
- Perform signed 6-bit binary addition of $-10 + (-13)$

Signed 2s Complement Subtraction

- Perform signed 6-bit binary addition of $8 - 11$
- Perform signed 6-bit binary addition of $-8 - 11$
- Perform signed 6-bit binary addition of $8 - (-11)$
- Perform signed 6-bit binary addition of $-8 - (-11)$

Signed Arithmetics Functions

- Signed 2s Complement Representation
- Signed Binary Addition and Subtraction

Other Arithmetics Functions

Incrementing, Decrementing, Multiplication and
Division, Zero Fill and Extension

1. Incrementing and Decrementing

- Incrementing: adding a fixed value to an arithmetic variable (usually 1)
- Decrementing: subtracting a fixed value from an arithmetic variable (usually 1)

2. Multiplication and Division By Constants

- Design of full multiplier and divider is not hard in theory, but quite laborious
 1. Truth table: 2 x n -bit inputs, 2^{2n} rows in truth table
Require automated design
 2. Use functional blocks
e.g. n adders stacked with n enablers
- Friends don't let friends do this by hand.

2. Multiplication and Division By Constants

- Design of full multiplier and divider is not hard in theory, but quite laborious
- By Constants
 - n -bit multiplied by (or divided by) m -bit, where $m < n$
e.g. 4-bit to 3-bit multiplier
 - easier to design by hand, can be used as functional blocks for full implementation

3. Zero Fill and Extension

- Shift: shifting bits left or right

$$(1)_2 \ll 3 == (1000)_2; (111)_2 \gg 2 == (1)_2;$$

- Zero Fill: prepend/append Zeroes

$$(1100)_2 \rightarrow (0000\ 1100)_2; (1100)_2 \rightarrow (1100\ 0000)_2;$$

- Extension (of signed 2s complement integer): prepend Zeroes or 1s without changing value

Positive: prepend zeros; Negative: prepend 1s;

Other Arithmetics Functions

1. Incrementing and Decrementing
2. Multiplication and Division by Constants
3. Zero Fill and Extension

Lecture 3 Review

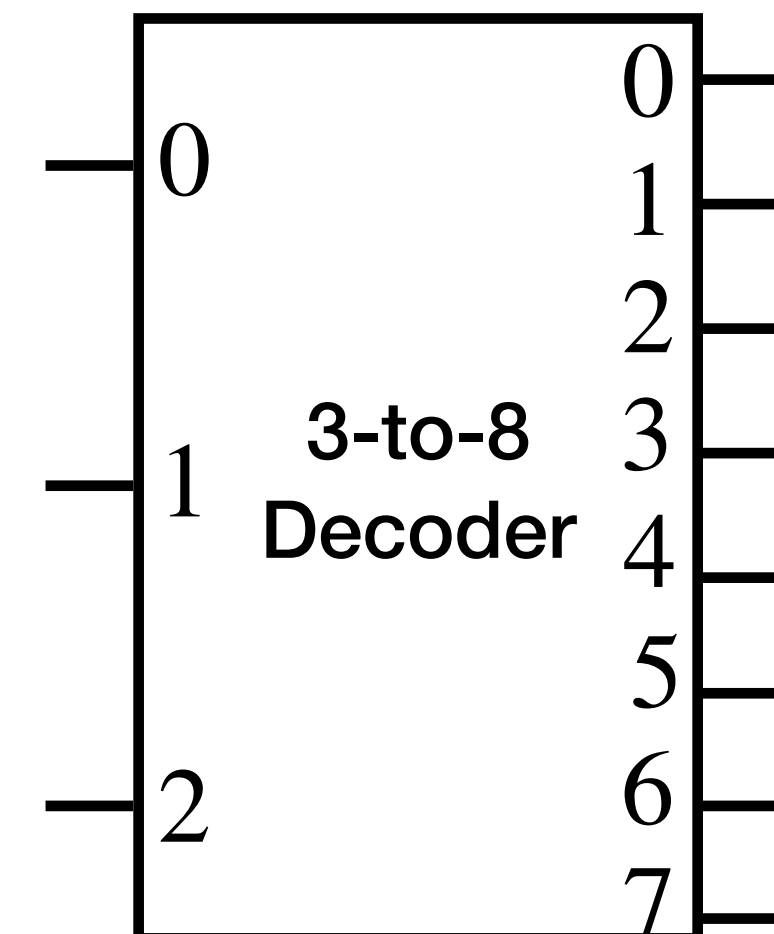
- Binary Adder: 1-bit Half Adder; 1-bit Full Adder; n -bit Adder
- Binary Subtractor: 1-bit Subtractor; n -bit subtractor
- 2s complement
- Binary Adder-Subtractor Unit
using Adder, Subtractor, Complementer and Multiplexer
- Binary Adder-Subtractor Unit
using Adder and XOR

Systematic Design Procedures

1. **Specification:** Write a specification for the circuit
2. **Formulation:** Derive relationship between inputs and outputs of the system e.g. using truth table or Boolean expressions
3. **Optimisation:** Apply optimisation, minimise the number of logic gates and literals required
4. **Technology Mapping:** Transform design to new diagram using available implementation technology
5. **Verification:** Verify the correctness of the final design in meeting the specifications

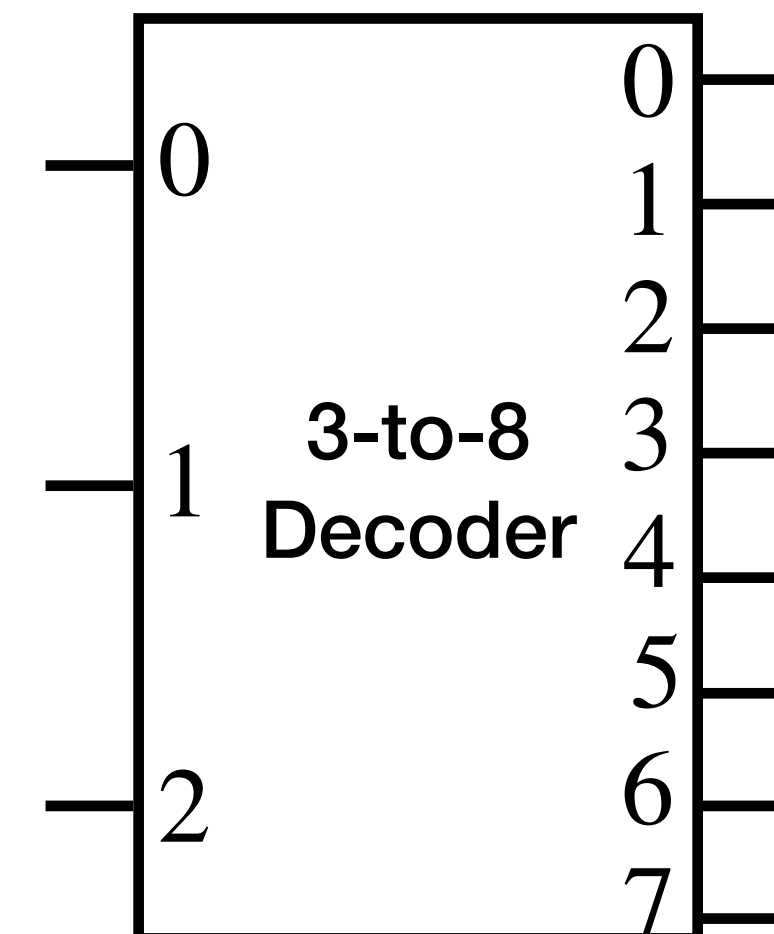
Functional Components (1)

- Value-Fixing, Transferring, Inverting, Enabler
- Decoder
 - Input: $A_0A_1 \dots A_{n-1}$
 - Output: $D_0D_1 \dots D_{2^n-1}$, $D_i = m_i$



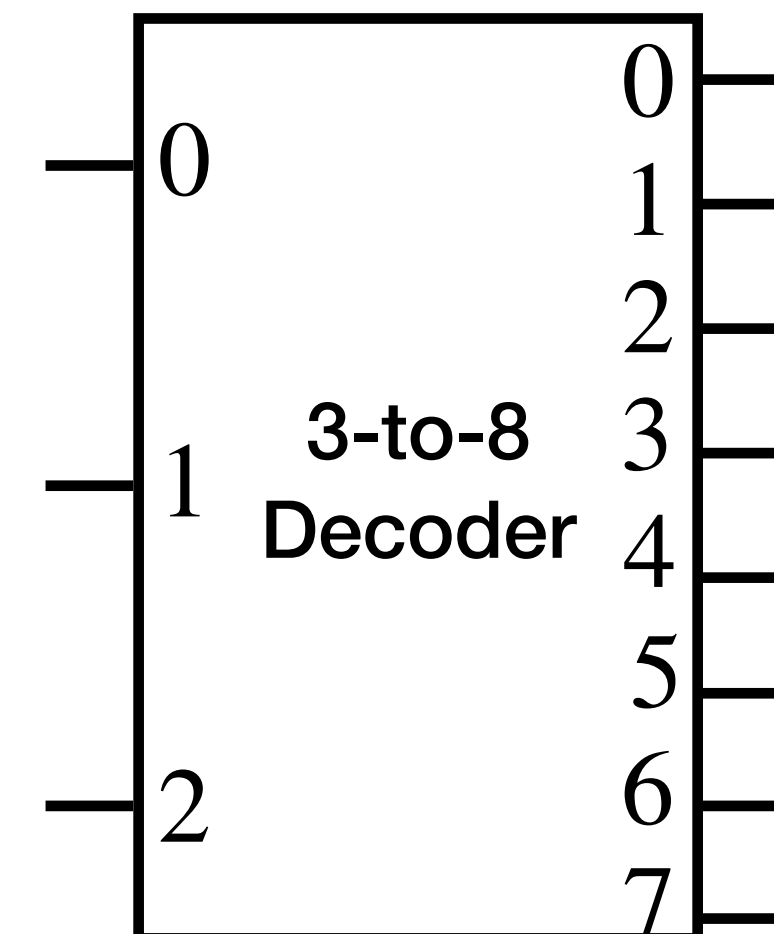
Functional Components (2)

- Encoder
 - Input: m_0, \dots, m_{2^n-1} with only one positive value
 - Output: A_0, \dots, A_{n-1}
 - Priority Encoder: validity, priority output
- Multiplexer
 - Switching between multiple input channels



Functional Components (2)

- Encoder
 - Input: m_0, \dots, m_{2^n-1} with only one positive value
 - Output: A_0, \dots, A_{n-1}
 - Priority Encoder: validity, priority output
- Multiplexer
 - Switching between multiple input channels



Arithmetic Units (Unsigned)

- Binary Adder: 1-bit Half Adder; 1-bit Full Adder; n -bit Adder
- Binary Subtractor: 1-bit Subtractor; n -bit subtractor
- 2s complement
- Binary Adder-Subtractor Unit
using Adder, Subtractor, Complementer and Multiplexer
- Binary Adder-Subtractor Unit
using Adder and XOR

Arithmetic Units

- Signed 2s Complement
- Signed Addition and Subtraction
- Incrementing and Decrementing
- Multiplication and Division by Constants
- Zero Fill and Extension

Today's Tasks

- 1-bit Half Adder
- 1-bit Full Adder using 1-bit Half Adder in Schema Diagram (Logic Circuit Diagram)
- 4-bit Full Adder using 1-bit Full Adder in Schema Diagram (Logic Circuit Diagram)
- 4-bit Adder-Subtractor using 4-bit Full Adder in Schema Diagram (Logic Circuit Diagram)