03.03.21 12:03

CSCI 150
Introduction to Digital and Computer System Design
Midterm Review II

Jetic Gū

Overview

- Focus: Review
- Architecture: Combinational Logic Circuit
- Textbook v4: Ch1-4; v5: Ch1-3
- Core Ideas:
    1. Digital Information Representation (Lecture 1)
    2. Combinational Logic Circuits (Lecture 2)
    3. Combinational Functional Blocks, Arithmetic Blocks (Lecture 3)

Lecture 3: Combinational Logic Design

5 Steps Systematic Design Procedures; Functional Blocks; Decoder, Enabler, Multiplexer; Arithmetic Blocks

Summary

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

Concept

Hierarchical Design

- "divide-and-conquer"
- Circuit is broken up into individual functional pieces (blocks)
  - Each block has explicitly defined **Interface** (I/O) and **Behaviour**
  - A single block can be **reused** multiple times to simplify design process
  - If a single block is too complex, it can be **further divided into smaller blocks**, to allow for easier designs

P3.2
Elementary Func.

Value-Fixing, Transferring, and Inverting

① **Value-Fixing**: giving a constant value to a wire

- $F = 0; F = 1;$

② **Transferring**: giving a variable (wire) value from another variable (wire)

- $F = X;$

③ **Inverting**: inverting the value of a variable

- $F = \overline{X}$

Concept

Vector Denotation

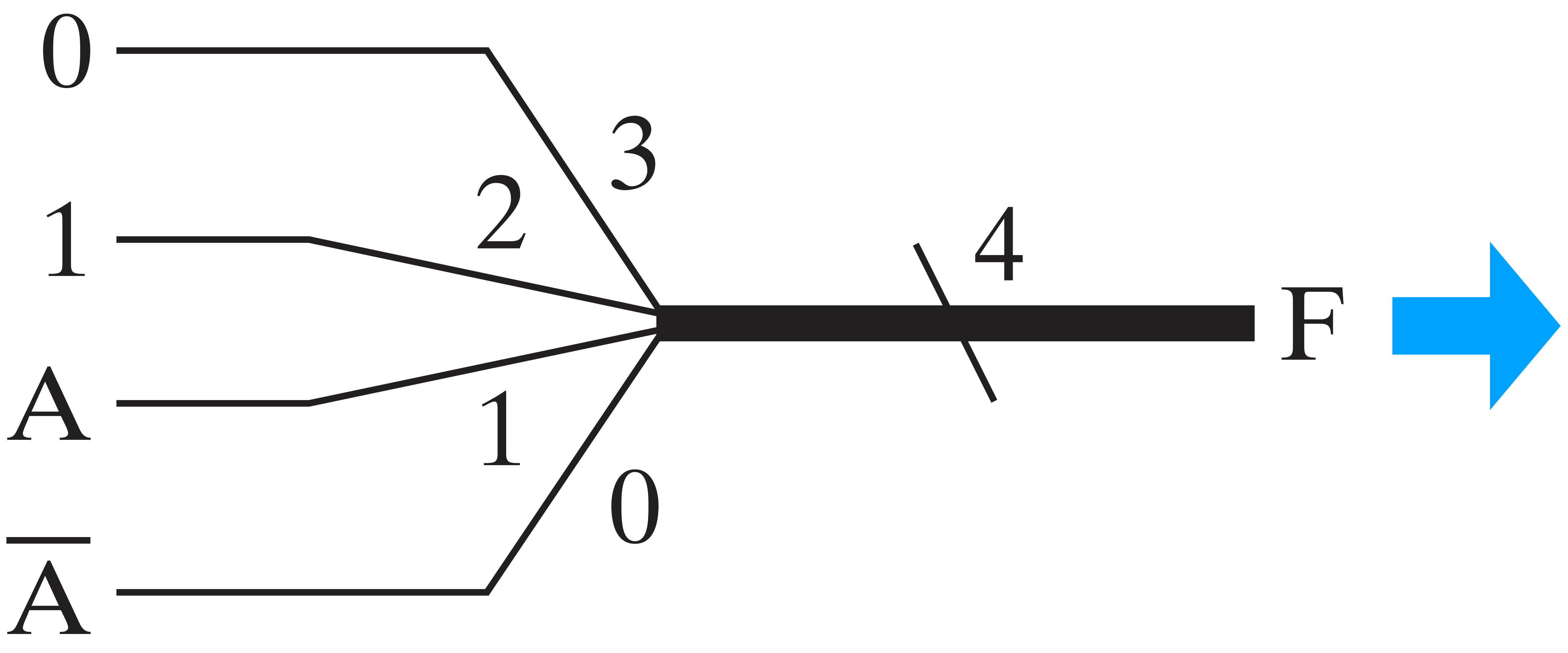
④ **Multiple-bit Function**

- Functions we've seen so far has only one-bit output: 0/1
- Certain functions may have n-bit output
  - $F(n-1:0) = (F_{n-1}, F_{n-2}, \ldots, F_0)$, each F_i is a one-bit function
  - Curtain Motor Control Circuit: $F = (F_{\text{Motor}_1}, F_{\text{Motor}_2}, F_{\text{Light}})$

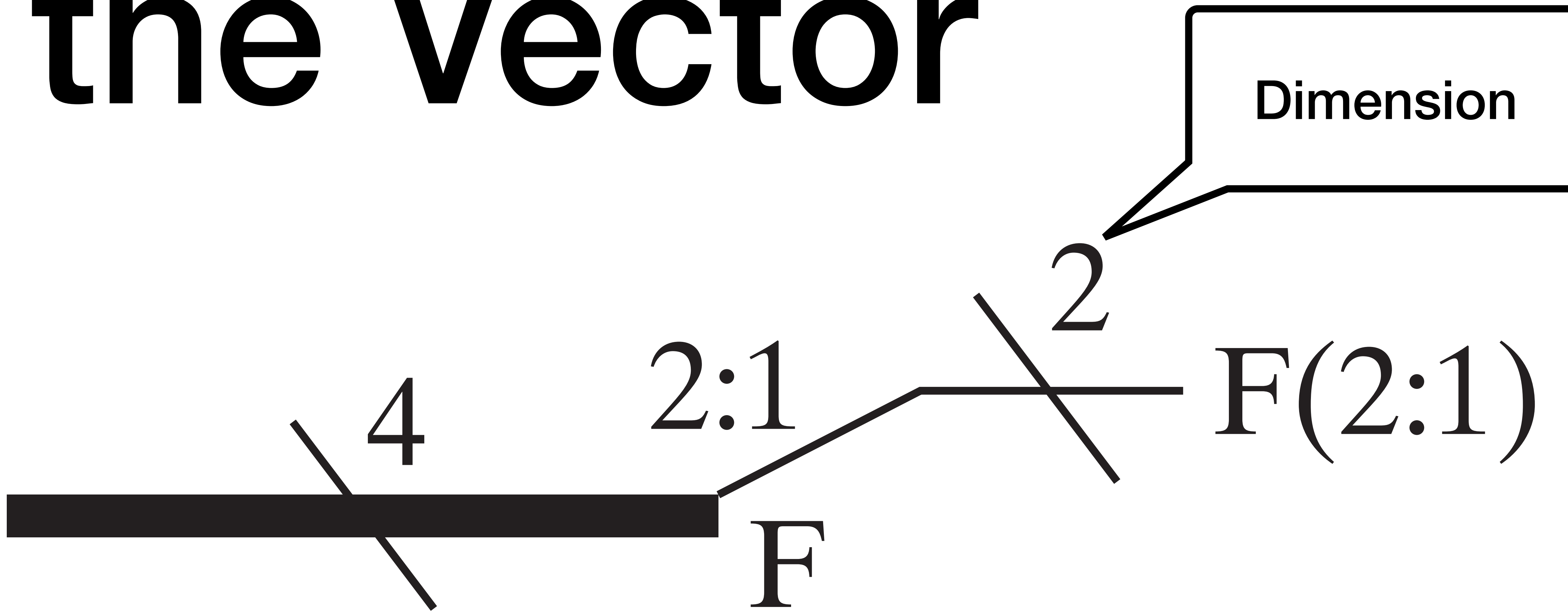
Concept

P1
Elementary Func.

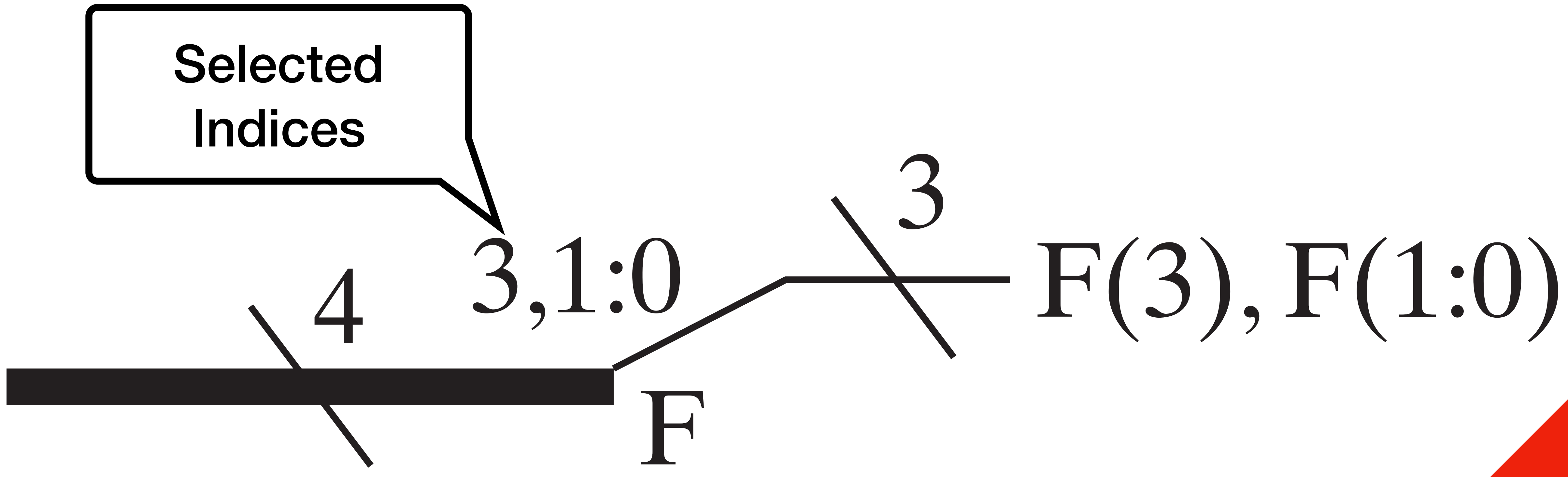
Taking part of the Vector



④ Multiple-bit Function



Output: (F_2, F_1)



Output: (F_3, F_1, F_0)

Concept

Enabler

⑤ Enabler

- Transferring function, but with an additional EN signal acting as switch

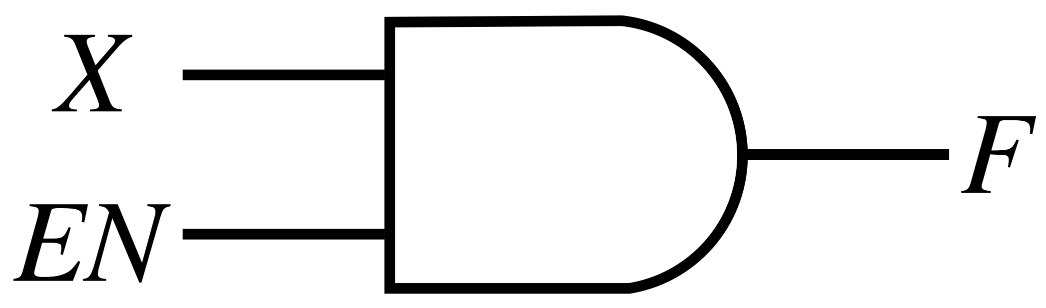
EN	X	F
0	X	0
1	0	0
1	1	1

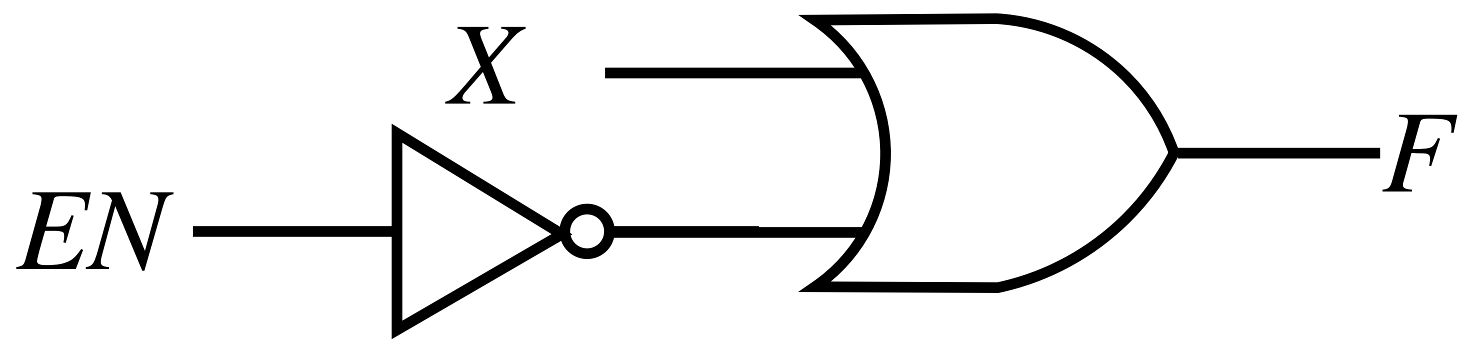
Concept

Enabler

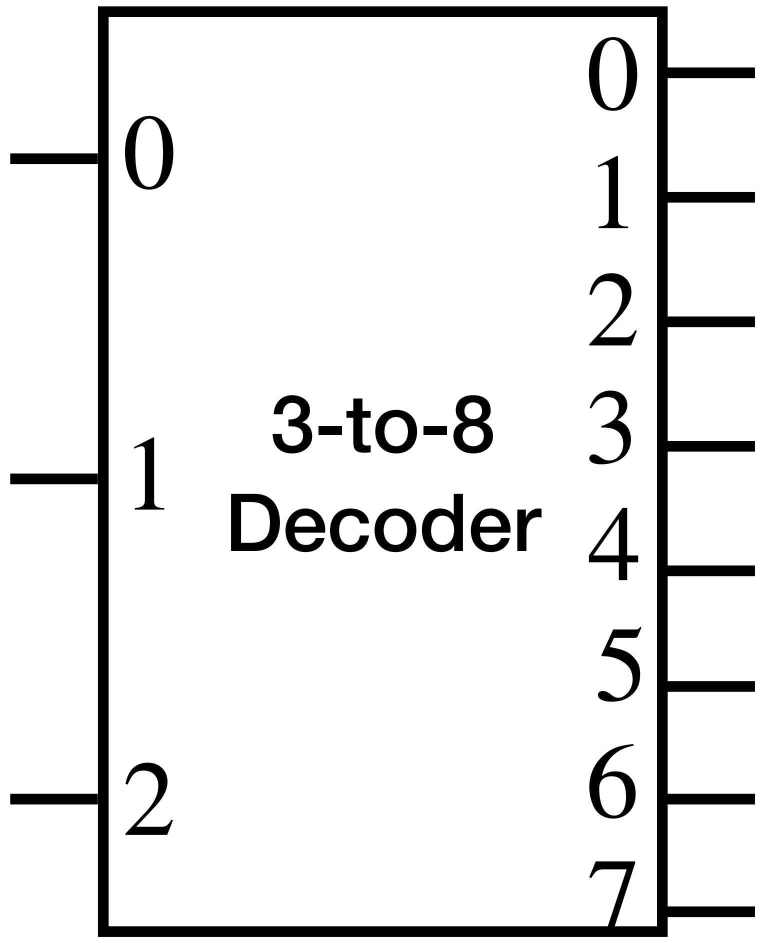
⑤ **Enabler**

- Transferring function, but with an additional EN signal acting as switch



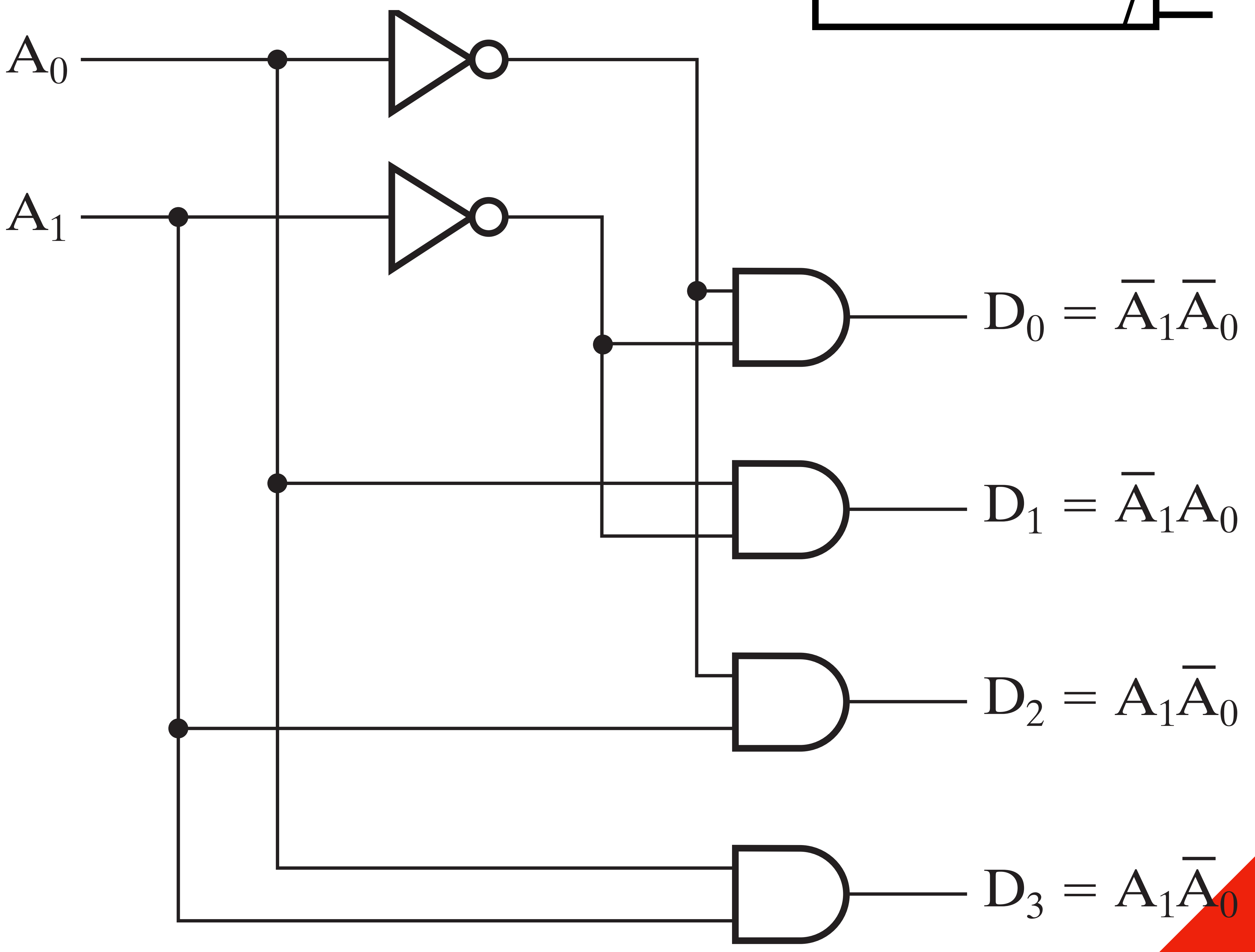


Decoder



- n-bit input, 2^nbits output
  - $D_i = m_i$
- Design: use hierarchical designs!

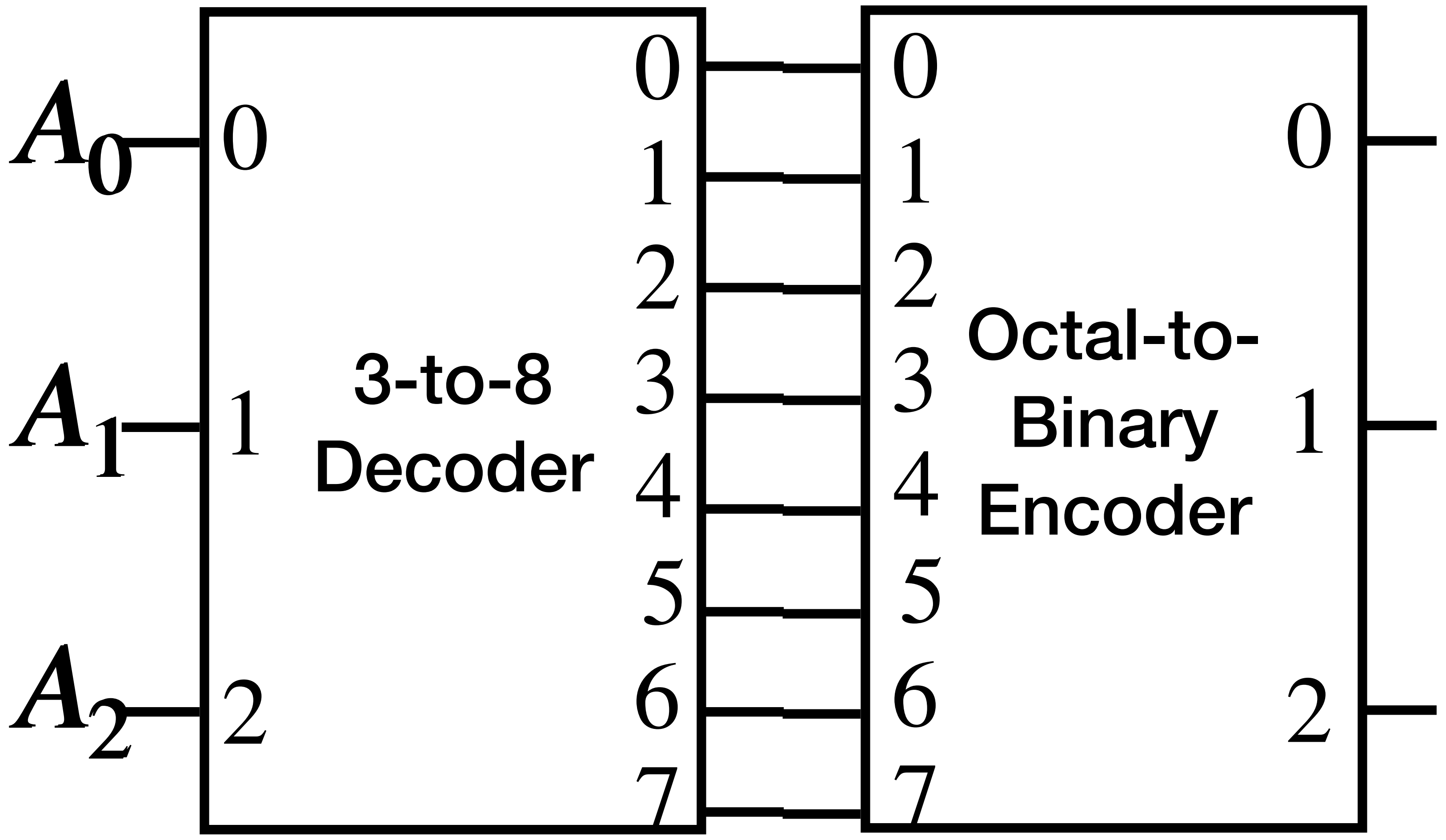
A_1	A_0	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



Concept

Encoder

- Inverse operation of a decoder
- 2^n inputs, only one is giving positive input[1]
- n outputs

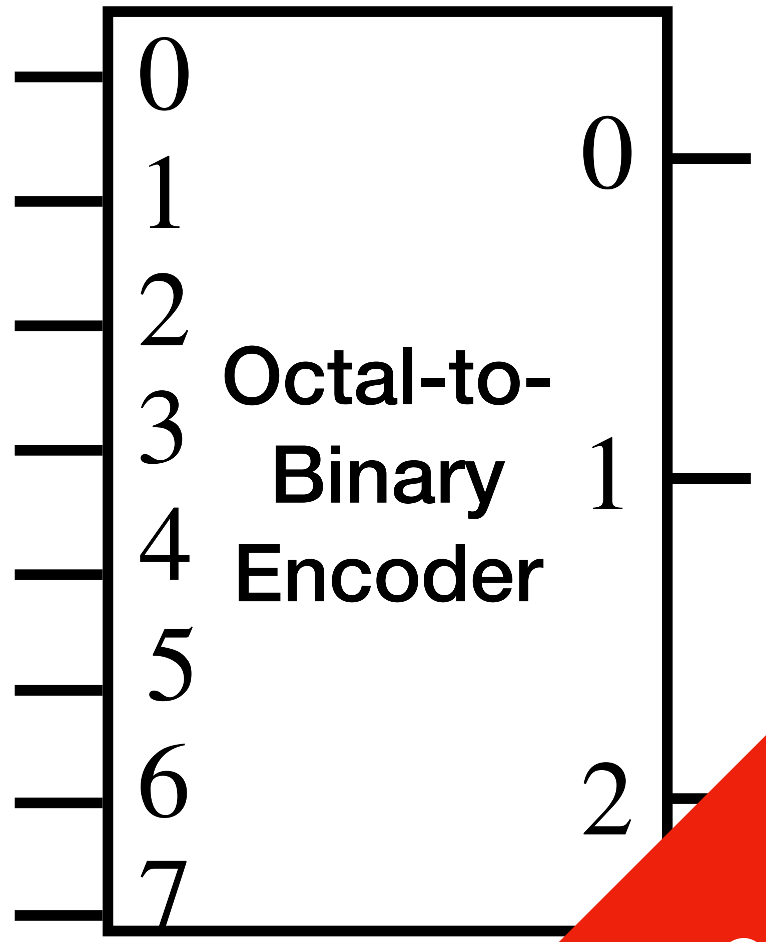


1. In reality, could be less

Encoder

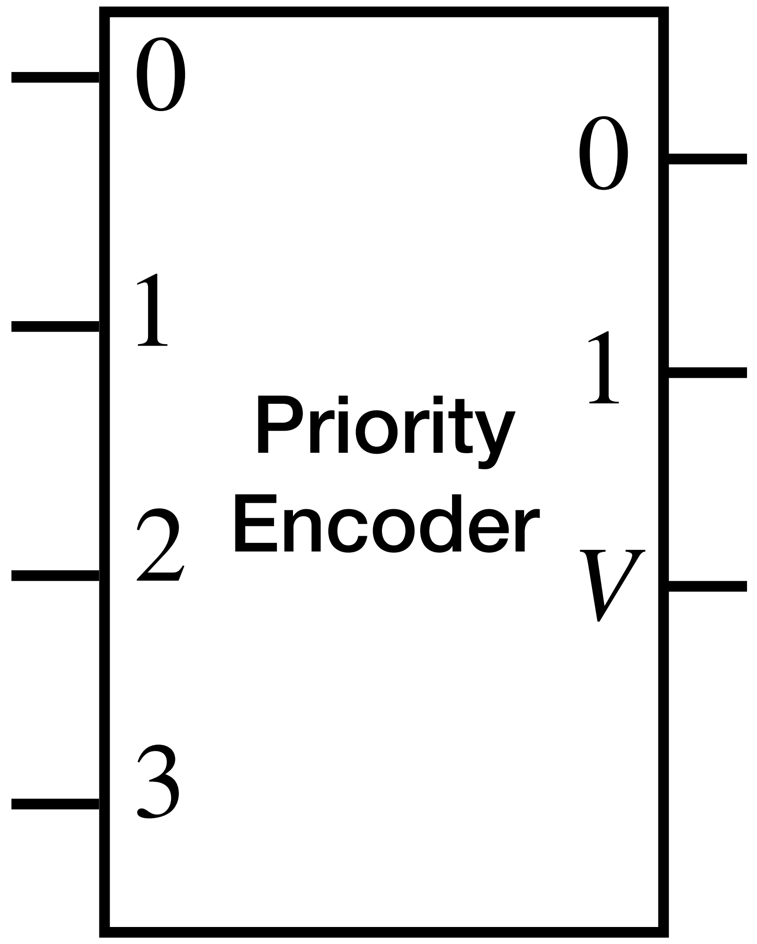
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0
							1	0	0	0
						1		0	0	1
					1			0	1	0
				1				0	1	1
			1					1	0	0
		1						1	0	1
	1							1	1	0
1								1	1	1

$$A_0 = D_1 + D_3 + D_5 + D_7$$
$$A_1 = D_2 + D_3 + D_6 + D_7$$
$$A_2 = D_4 + D_5 + D_6 + D_7$$



Priority Encoder

- Additional Validity Output V
  - Indicating whether the input is valid (contains 1)
- Priority
  - Ignores $D_{<i}$ if $D_i = 1$



P3.3
Adv. Func. Blocks

Priority Encoder

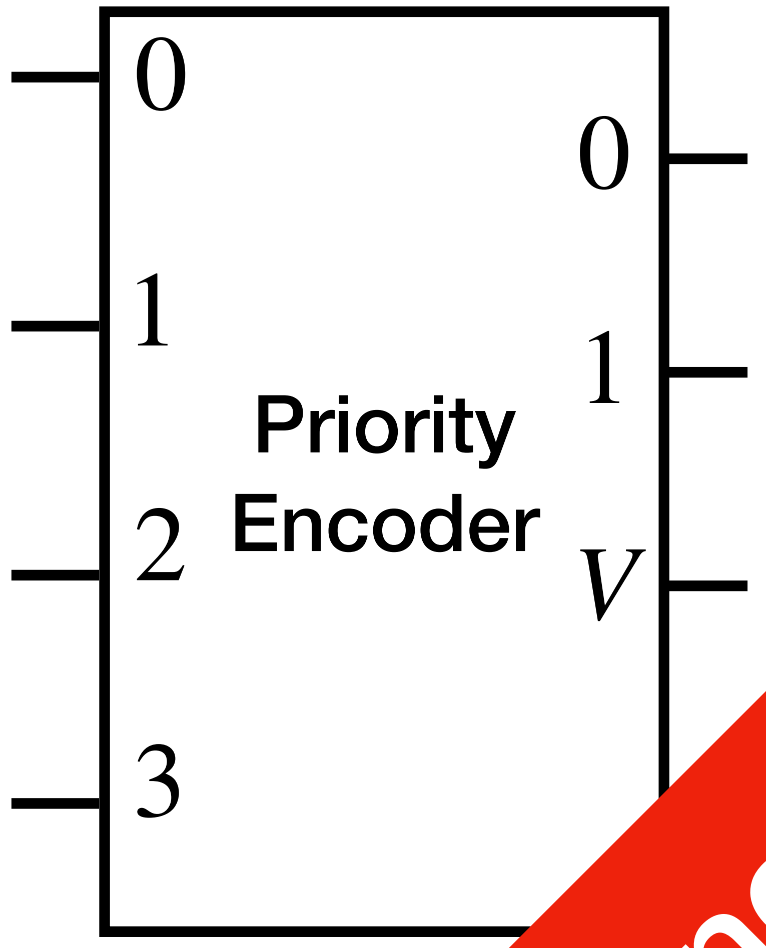
D_3	D_2	D_1	D_0	A_1	A_0	V
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

$$V = D_3 + D_2 + D_1 + D_0$$

$$A_1 = D_3 + \overline{D_3}D_2 = D_2 + D_3$$

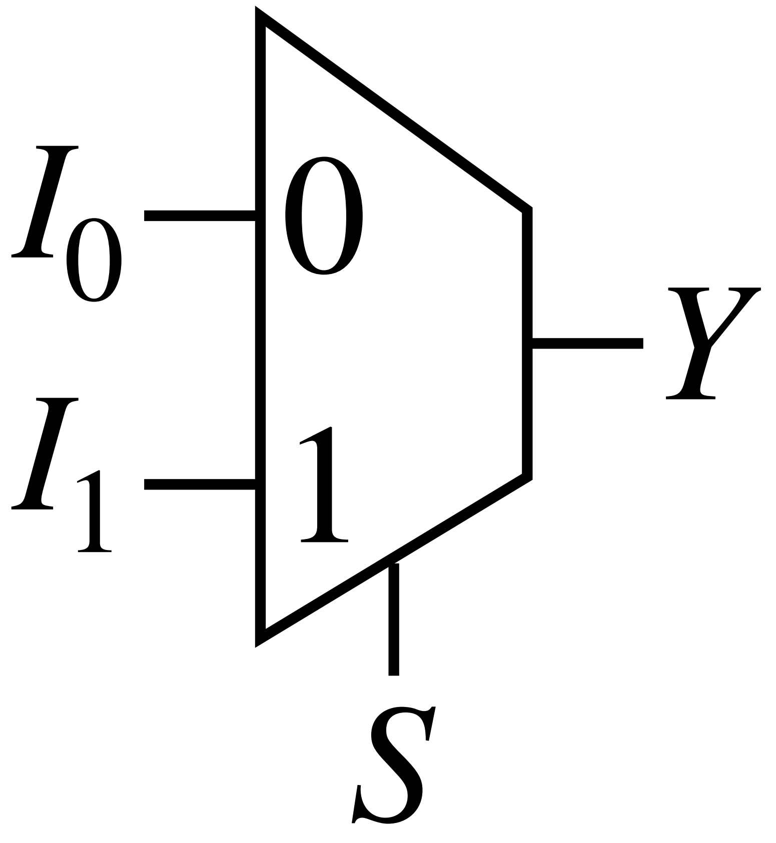
$$A_0 = \overline{D_3}\,\overline{D_2}D_1 + D_3$$

$$= \overline{D_2}D_1 + D_3$$

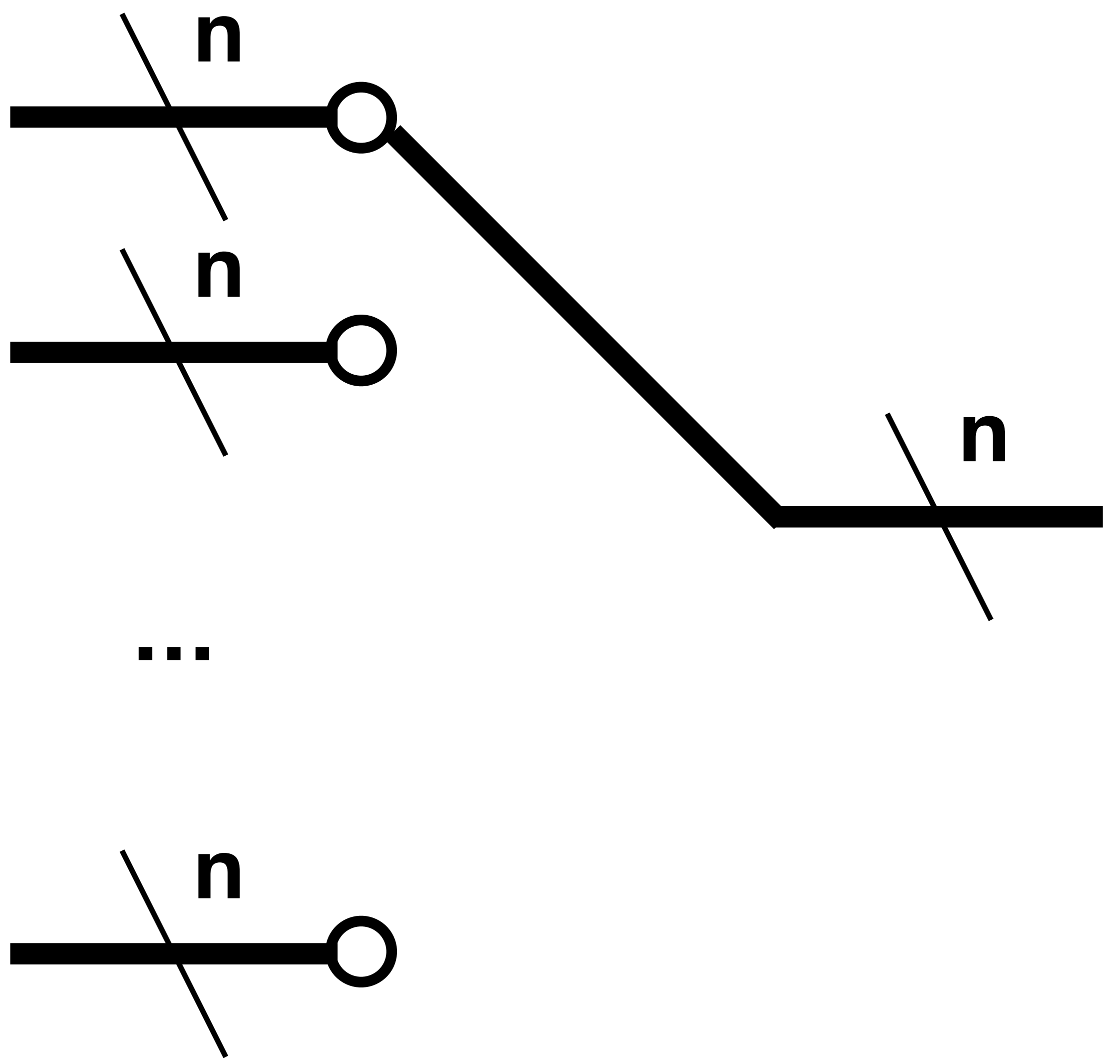


Concept

Multiplexer



- Multiple n-variable input vectors
- Single n-variable output vector
- Switches: which input vectors to output



Concept

Common Techniques

- Implementing Multiplexer using Decoders
- Implementing Multiplexer using smaller Multiplexers
- Implementing Sum-of-Minterm using Decoder
  (use OR gate)
- Implementing Sum-of-Minterm using Multiplexer
  (use value fixing)

Concept

Arithmetic Blocks

- 1-bit Half Adder and Full Adder
- n-bit Adder
- 1-bit subtractor and n-bit subtractor
- 2s complement and binary adder-subtractor

Concept

1-bit Adder

	Half adder	Full adder
Carries	C	$C\ Z$
Augend	X	X
Addend	$+Y$	$+Y$
Sum	S	S

Input
Output

- Half adder
  input X, Y
  output S, C

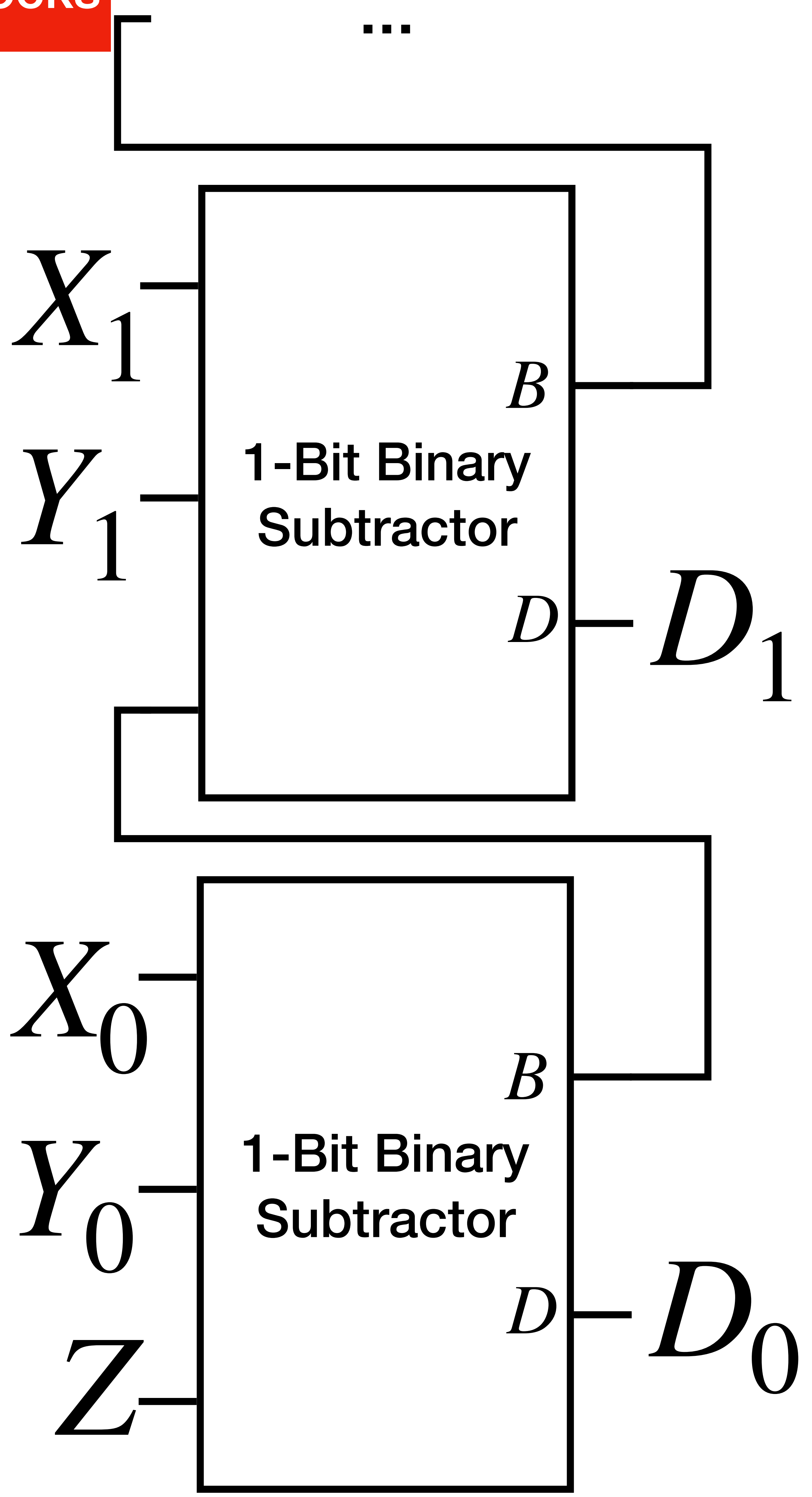
- Full adder
  input X, Y, Z;
  output S, C

Review

P3.4
Arithmetic Blocks

Unsigned Binary Subtraction

Technology
- 1 bit Unsigned Subtractor

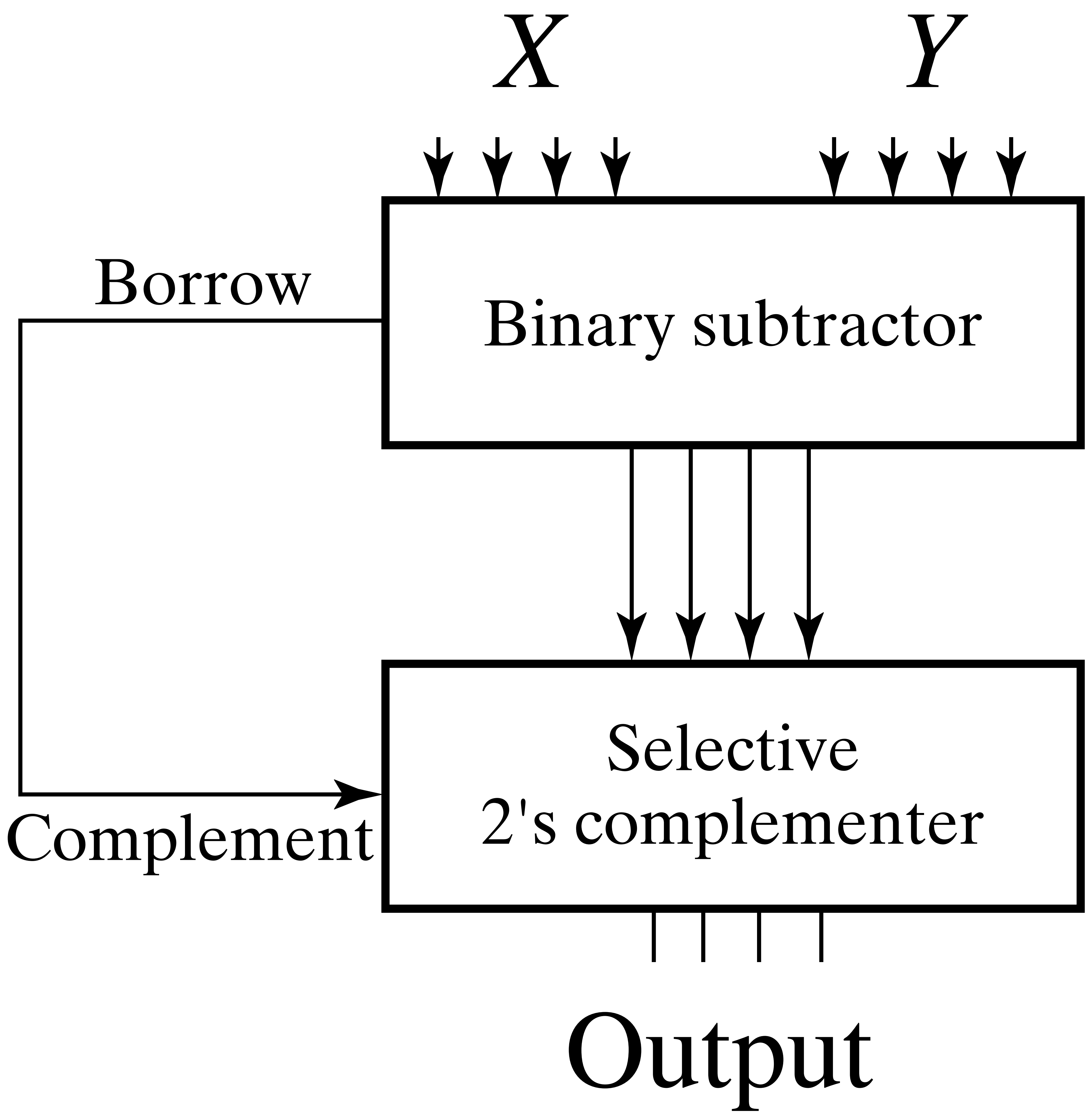


	B ... Z
Borrows	000110
Minuend $X_{0:n-1}$	10110
Subtrahend $Y_{0:n-1}$	− 10011
Difference $D_{0:n-1}$	00011

Input
Output

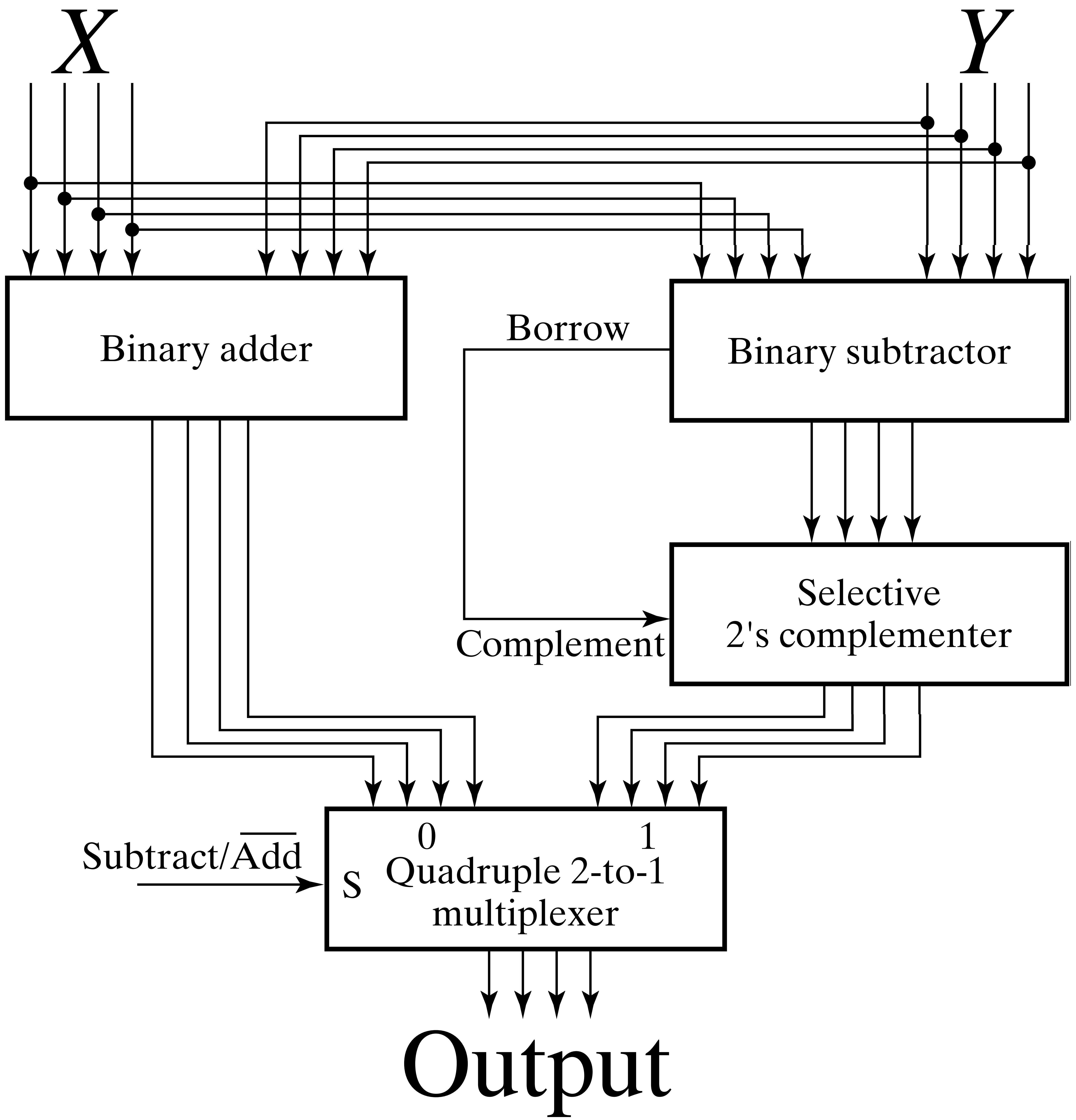
Review

Full Unsigned Subtraction



Concept

Adder Subtractor I



Adder-Subtractor II