13.06.20 11:57

CSCI 150
Introduction to Digital and Computer System Design
Lecture 3: Combinational Logic Design II

Jetic Gū

2020 Summer Semester (S2)

Overview

- Focus: Methodology
- Architecture: Combinatory Logical Circuits
- Textbook v4: Ch3 3.1, 3.2, 3.3; v5: Ch3 3.1, 3.2
- Core Ideas:
  1. BCD-to-Seven-Segment Decoder
  2. 4-bit Equity Comparator
  3. Technology Mapping

P0
Review

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

Review

P1
Example 3b

BCD-to-Seven-Segment Decoder

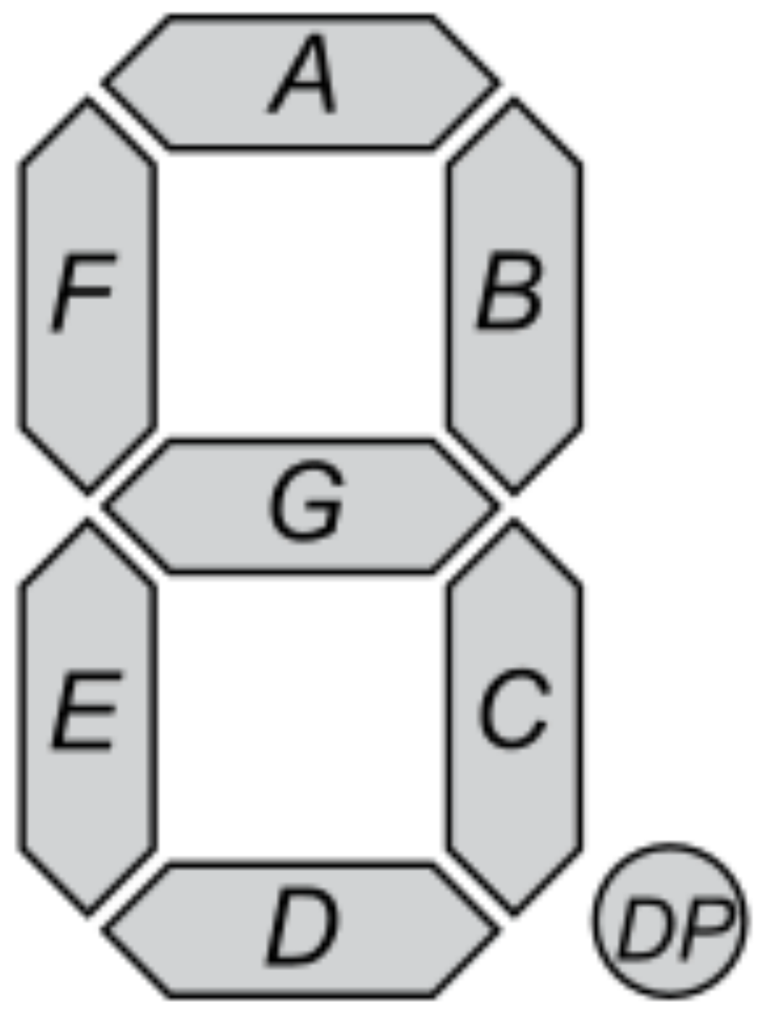
Ah, not again

Summary

P1
Example 3b

LED Seven Segment Display

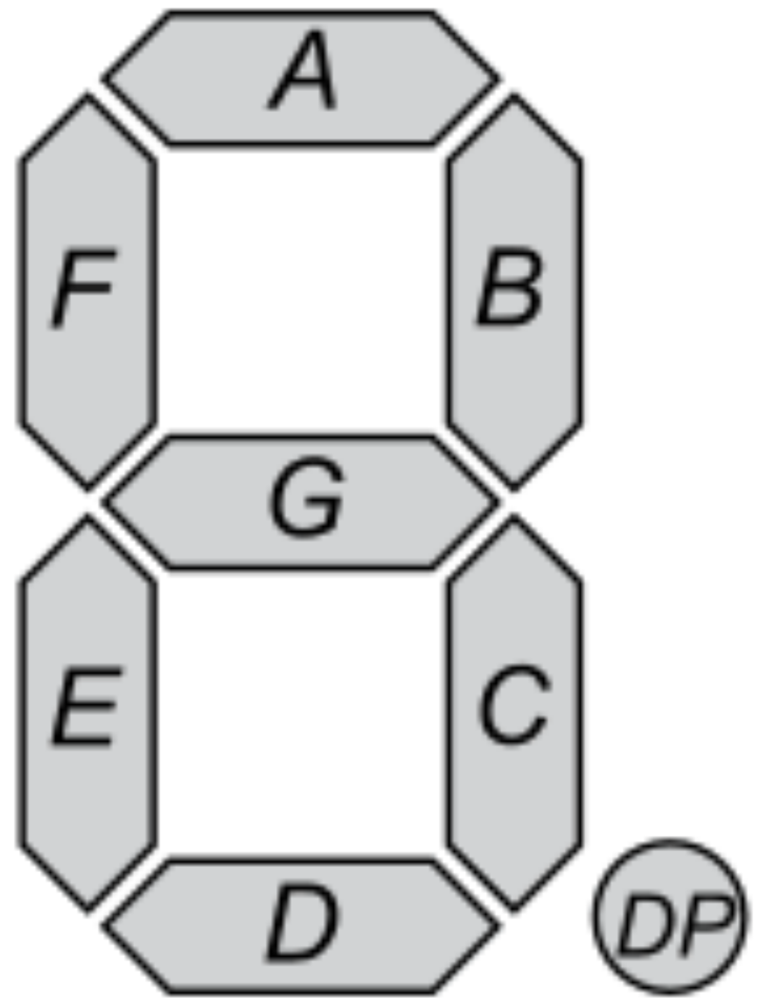
- LED: Light-Emitting Diodes
- A single digit display takes 7bit inputs (and an optional one for decimal point)



Example

BCD to 7 Segment Display(s)

- BCD: Each digit is represented using 4bit binary int
- BCD-to-7-segment decoder
  A Combinational circuit that
  - takes a decimal digit in BCD (4bit int A, B, C, D); and
  - generates the appropriate control signals for the display unit (a, b, c, d, e, f, g)



P1
Example 3b

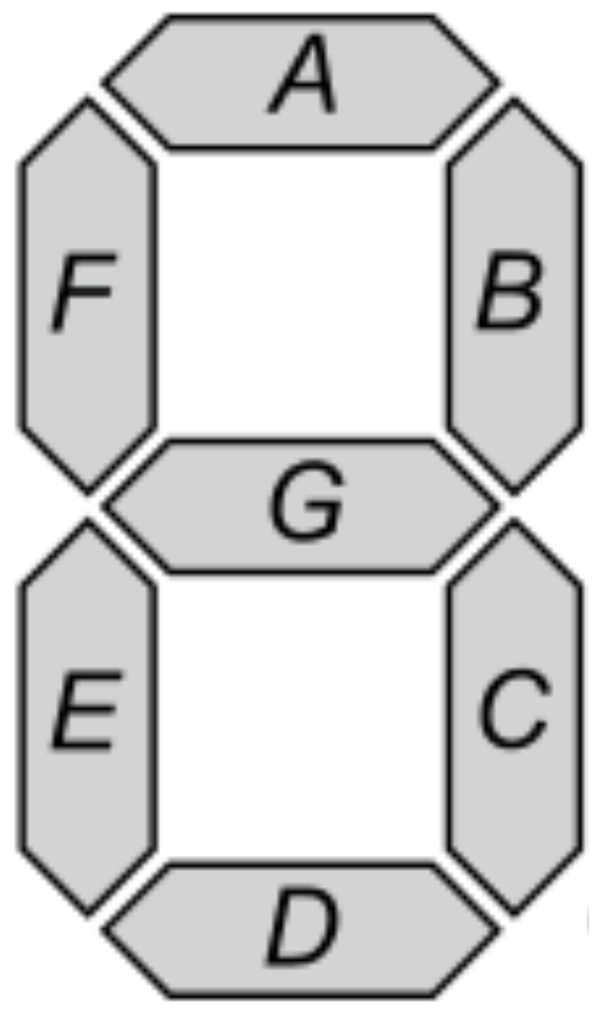
1. Specification

- BCD: Each digit is represented using 4bit binary int
- BCD-to-7-segment decoder
  A Combinational circuit that

Input
- takes a decimal digit in BCD (4bit int A, B, C, D); and

Output
- generates the appropriate control signals for the display unit (a, b, c, d, e, f, g)

Output



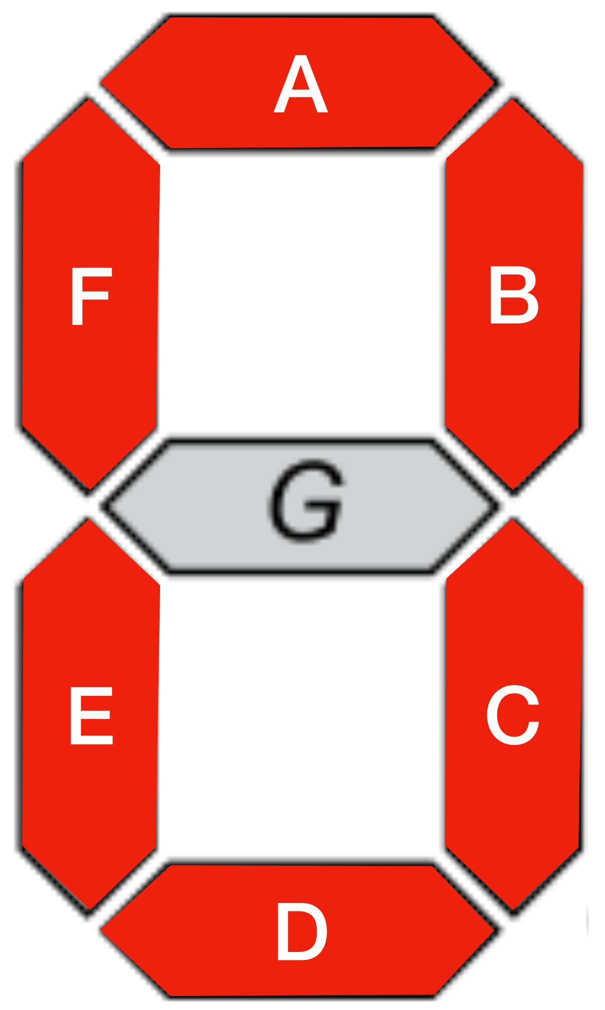
Example

P1
Example 3b

2. Formulation

Decimal	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	
1											
2											
3											
4											
5											
6											
7											
8											
9											

Output



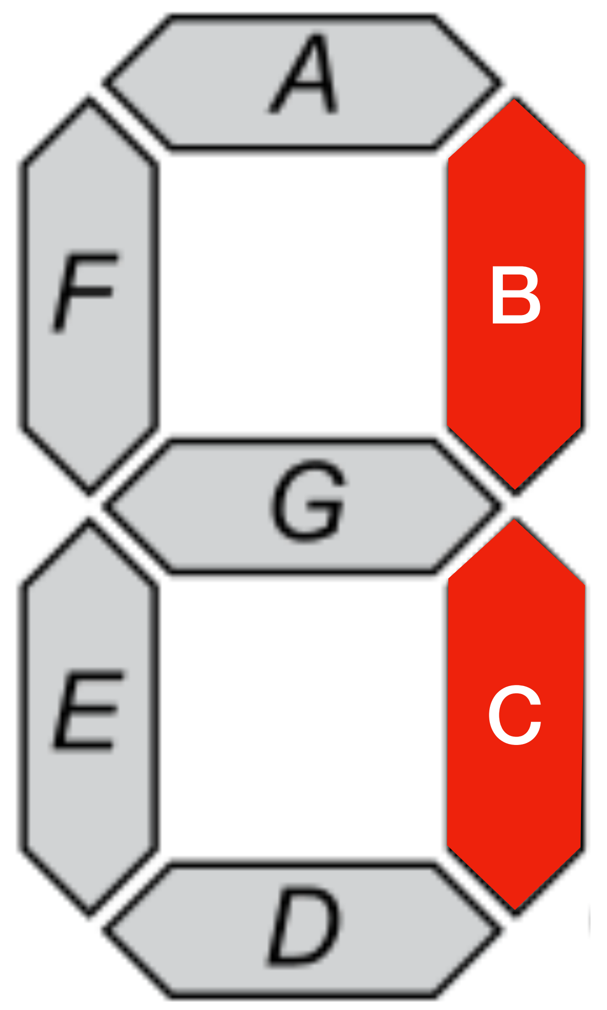
Example

P1
Example 3b

2. Formulation

Decimal	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	
1	0	0	0	1		1	1				
2											
3											
4											
5											
6											
7											
8											
9											

Output



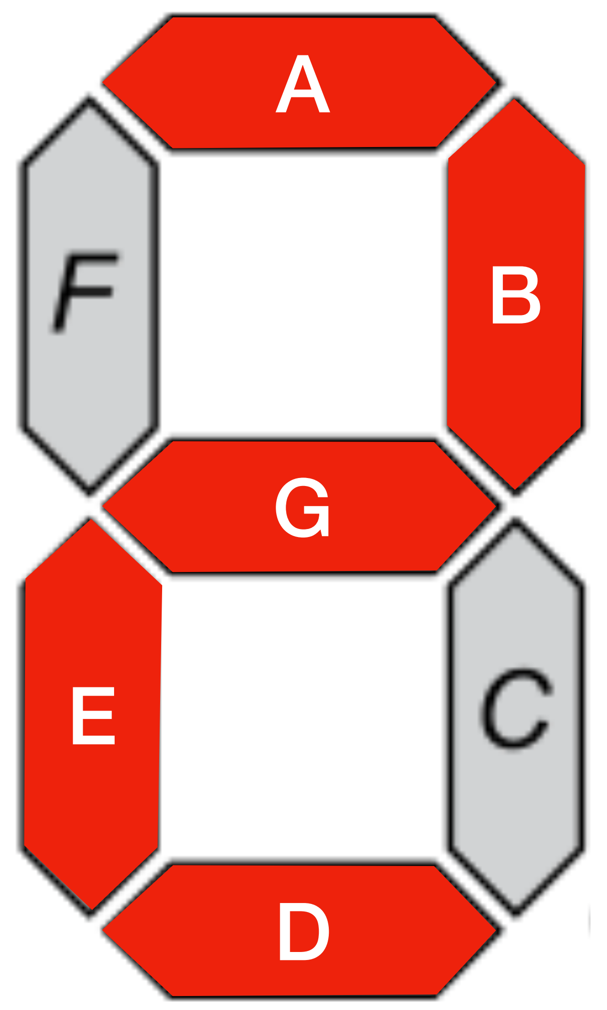
Example

P1
Example 3b

2. Formulation

Decimal	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	
1	0	0	0	1		1	1				
2	0	0	1	0	1	1		1	1		1
3											
4											
5											
6											
7											
8											
9											

Output



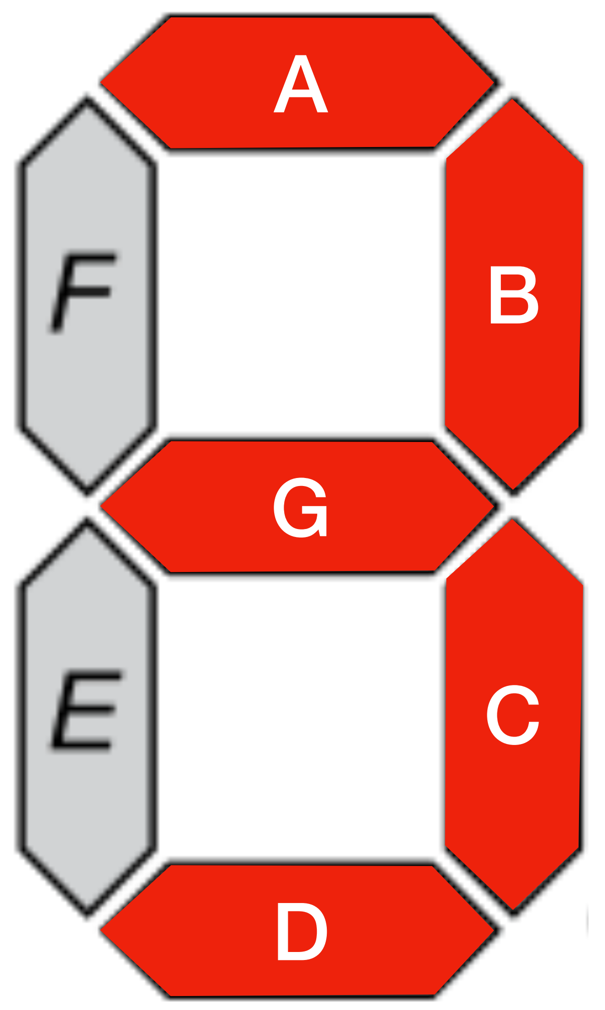
Example

P1
Example 3b

2. Formulation

Decimal	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	
1	0	0	0	1		1	1				
2	0	0	1	0	1	1		1	1		1
3	0	0	1	1	1	1	1	1			1
4											
5											
6											
7											
8											
9											

Output



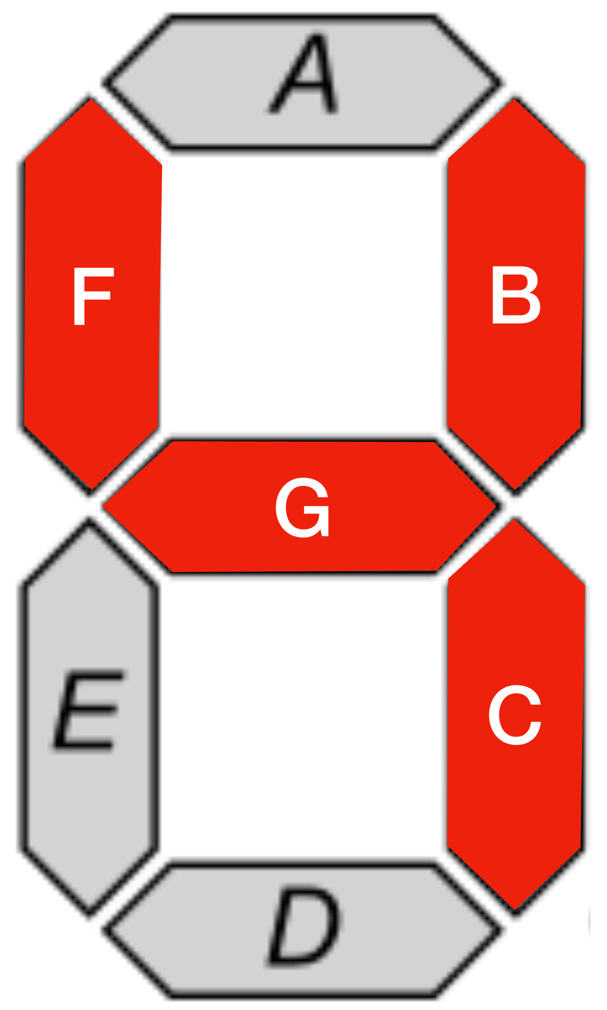
Example

P1
Example 3b

2. Formulation

Decimal	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	
1	0	0	0	1		1	1				
2	0	0	1	0	1	1		1	1		1
3	0	0	1	1	1	1	1	1			1
4	0	1	0	0		1	1			1	1
5											
6											
7											
8											
9											

Output



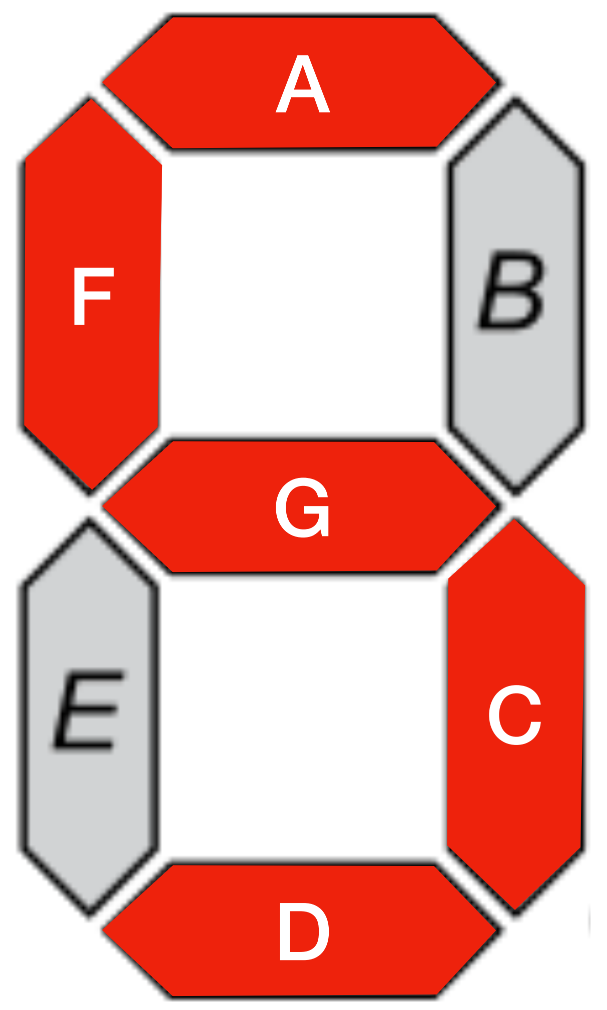
Example

P1
Example 3b

2. Formulation

Decimal	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	
1	0	0	0	1		1	1				
2	0	0	1	0	1	1		1	1		1
3	0	0	1	1	1	1	1	1			1
4	0	1	0	0		1	1			1	1
5	0	1	0	1	1		1	1		1	1
6											
7											
8											
9											

Output



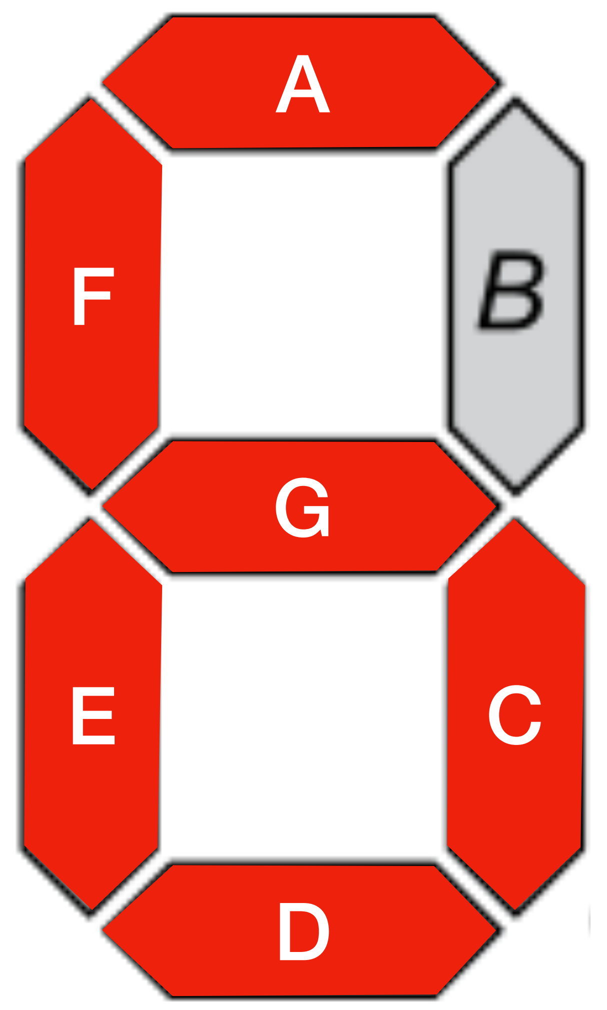
Example

P1
Example 3b

2. Formulation

Decimal	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	
1	0	0	0	1		1	1				
2	0	0	1	0	1	1		1	1		1
3	0	0	1	1	1	1	1	1			1
4	0	1	0	0		1	1			1	1
5	0	1	0	1	1		1	1		1	1
6	0	1	1	0	1		1	1	1	1	1
7											
8											
9											

Output



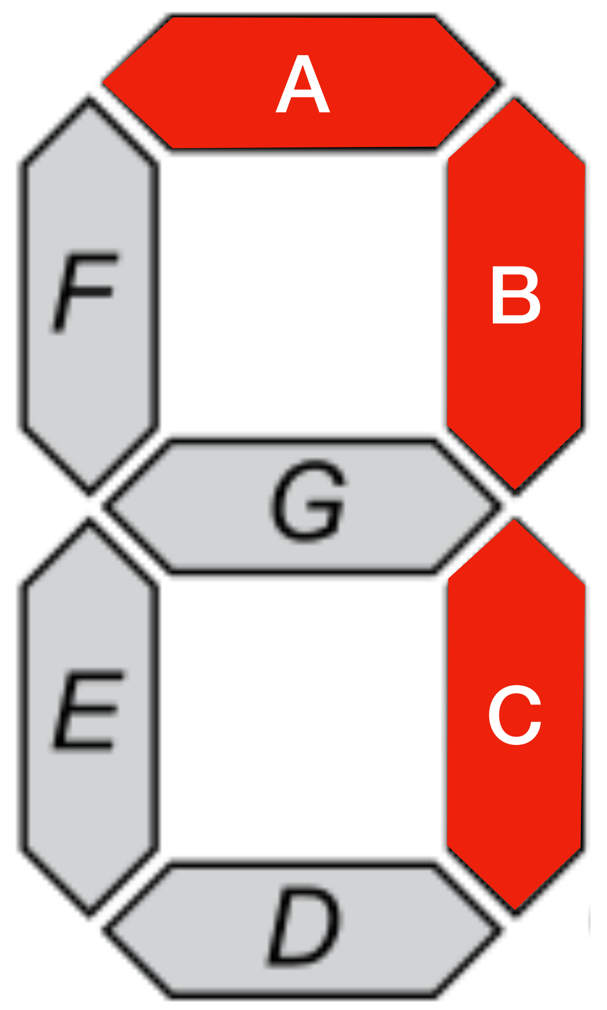
Example

P1
Example 3b

2. Formulation

Decimal	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	
1	0	0	0	1		1	1				
2	0	0	1	0	1	1		1	1		1
3	0	0	1	1	1	1	1	1			1
4	0	1	0	0		1	1			1	1
5	0	1	0	1	1		1	1		1	1
6	0	1	1	0	1		1	1	1	1	1
7	0	1	1	1	1	1	1				
8											
9											

Output



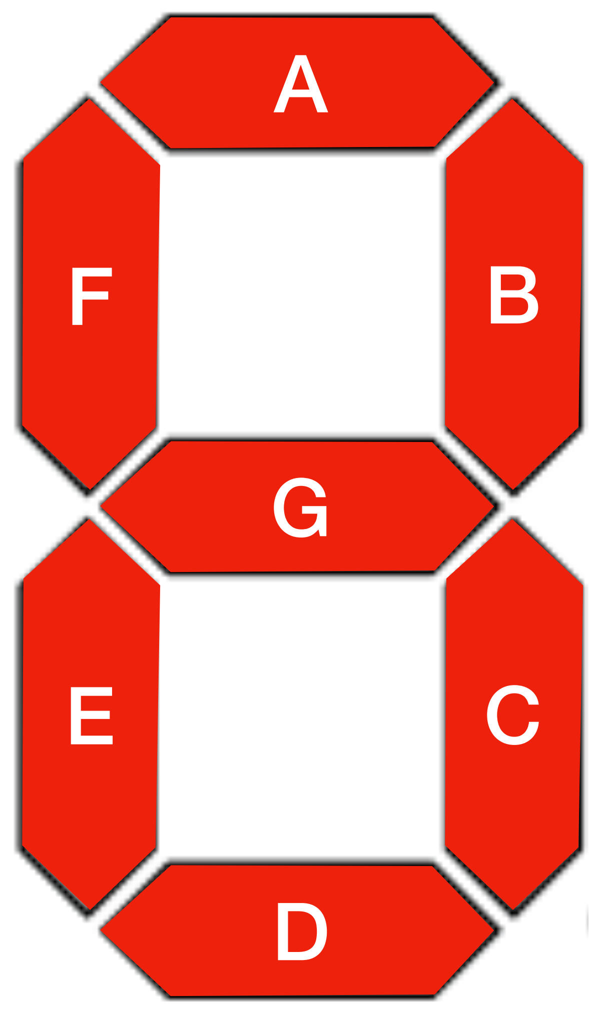
Example

P1
Example 3b

2. Formulation

Decimal	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	
1	0	0	0	1		1	1				
2	0	0	1	0	1	1		1	1		1
3	0	0	1	1	1	1	1	1			1
4	0	1	0	0		1	1			1	1
5	0	1	0	1	1		1	1		1	1
6	0	1	1	0	1		1	1	1	1	1
7	0	1	1	1	1	1	1				
8	1	0	0	0	1	1	1	1	1	1	1
9											

Output



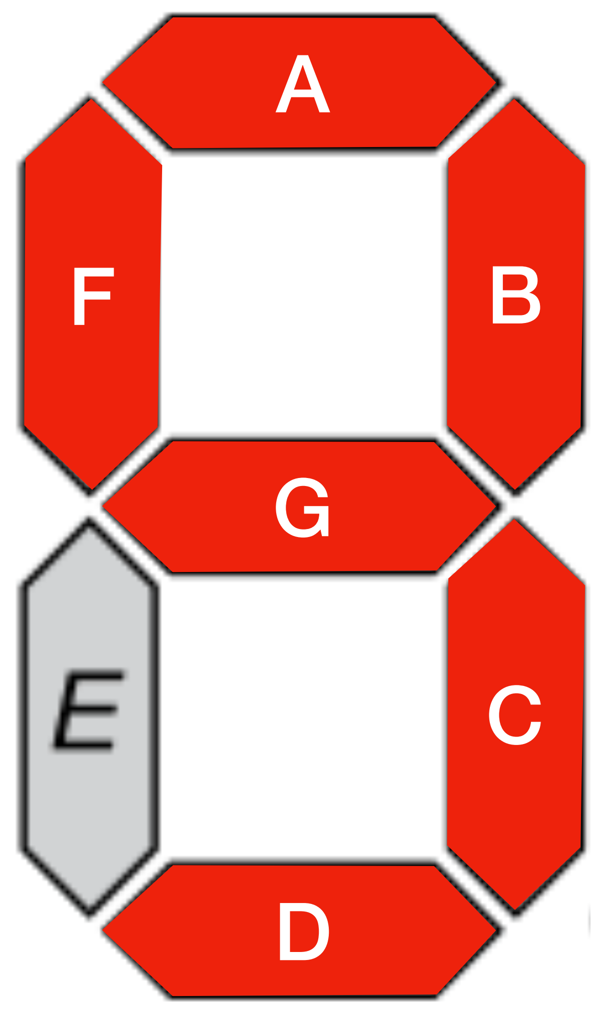
Example

P1
Example 3b

2. Formulation

Decimal	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	
1	0	0	0	1		1	1				
2	0	0	1	0	1	1		1	1		1
3	0	0	1	1	1	1	1	1			1
4	0	1	0	0		1	1			1	1
5	0	1	0	1	1		1	1		1	1
6	0	1	1	0	1		1	1	1	1	1
7	0	1	1	1	1	1	1				
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1		1	1

Output



Example

P1
Example 3b

3. Optimisation

- Convert to Boolean expression

Decimal	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

Example

P1
Example 3b

3. Optimisation

- Convert to Boolean expression

$$a = \overline{A}C + \overline{A}BD + \overline{B}\,\overline{C}\,\overline{D} + A\overline{B}\,\overline{C}$$

$$b = \overline{A}\,\overline{B} + \overline{A}\,\overline{C}\,\overline{D} + \overline{A}CD + A\overline{B}\,\overline{C}$$

$$c = \overline{A}B + \overline{A}D + \overline{B}\,\overline{C}\,\overline{D} + A\overline{B}\,\overline{C}$$

$$d = \overline{A}C\overline{D} + \overline{A}\,\overline{B}C + \overline{B}\,\overline{C}\,\overline{D} + A\overline{B}\,\overline{C} + \overline{A}B\overline{C}D$$

$$e = \overline{A}C\overline{D} + \overline{B}\,\overline{C}\,\overline{D}$$

$$f = \overline{A}B\overline{C} + \overline{A}\,\overline{C}\,\overline{D} + \overline{A}B\overline{D} + A\overline{B}\,\overline{C}$$

$$g = \overline{A}C\overline{D} + \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\,\overline{C}$$

Example

P1
Example 3b

4. Technology Mapping

- Convert to Boolean expression

- Indie: 27 AND
  7 OR gates

$$a = \overline{A}C + \overline{A}BD + \boxed{\overline{B}\,\overline{C}\,\overline{D}} + \boxed{A\overline{B}\,\overline{C}}$$

$$b = \overline{A}\,\overline{B} + \boxed{\overline{A}\,\overline{C}\,\overline{D}} + \overline{A}CD + \boxed{A\overline{B}\,\overline{C}}$$

$$c = \overline{A}B + \overline{A}D + \boxed{\overline{B}\,\overline{C}\,\overline{D}} + \boxed{A\overline{B}\,\overline{C}}$$

$$d = \boxed{\overline{A}C\overline{D}} + \boxed{\overline{A}\,\overline{B}C} + \boxed{\overline{B}\,\overline{C}\,\overline{D}} + \boxed{A\overline{B}\,\overline{C}} + \overline{A}B\overline{C}D$$

$$e = \boxed{\overline{A}C\overline{D}} + \boxed{\overline{B}\,\overline{C}\,\overline{D}}$$

$$f = \boxed{\overline{A}B\overline{C}} + \boxed{\overline{A}\,\overline{C}\,\overline{D}} + \overline{A}B\overline{D} + \boxed{A\overline{B}\,\overline{C}}$$

$$g = \boxed{\overline{A}C\overline{D}} + \boxed{\overline{A}\,\overline{B}C} + \boxed{\overline{A}B\overline{C}} + \boxed{A\overline{B}\,\overline{C}}$$

Example

P1
Example 3b

5. Verification (Skipped here)

- Convert to Boolean expression
- Indie: 27 AND
  7 OR gates

$$a = \overline{A}C + \overline{A}BD + \boxed{\overline{B}\,\overline{C}\,\overline{D}} + \boxed{A\overline{B}\,\overline{C}}$$

$$b = \overline{A}\,\overline{B} + \boxed{\overline{A}\,\overline{C}\,\overline{D}} + \overline{A}CD + \boxed{A\overline{B}\,\overline{C}}$$

$$c = \overline{A}B + \overline{A}D + \boxed{\overline{B}\,\overline{C}\,\overline{D}} + \boxed{A\overline{B}\,\overline{C}}$$

$$d = \boxed{\overline{A}C\overline{D}} + \boxed{\overline{A}\,\overline{B}C} + \boxed{\overline{B}\,\overline{C}\,\overline{D}} + \boxed{A\overline{B}\,\overline{C}} + \overline{A}B\overline{C}D$$

$$e = \boxed{\overline{A}C\overline{D}} + \boxed{\overline{B}\,\overline{C}\,\overline{D}}$$

$$f = \boxed{\overline{A}B\overline{C}} + \boxed{\overline{A}\,\overline{C}\,\overline{D}} + \overline{A}B\overline{D} + \boxed{A\overline{B}\,\overline{C}}$$

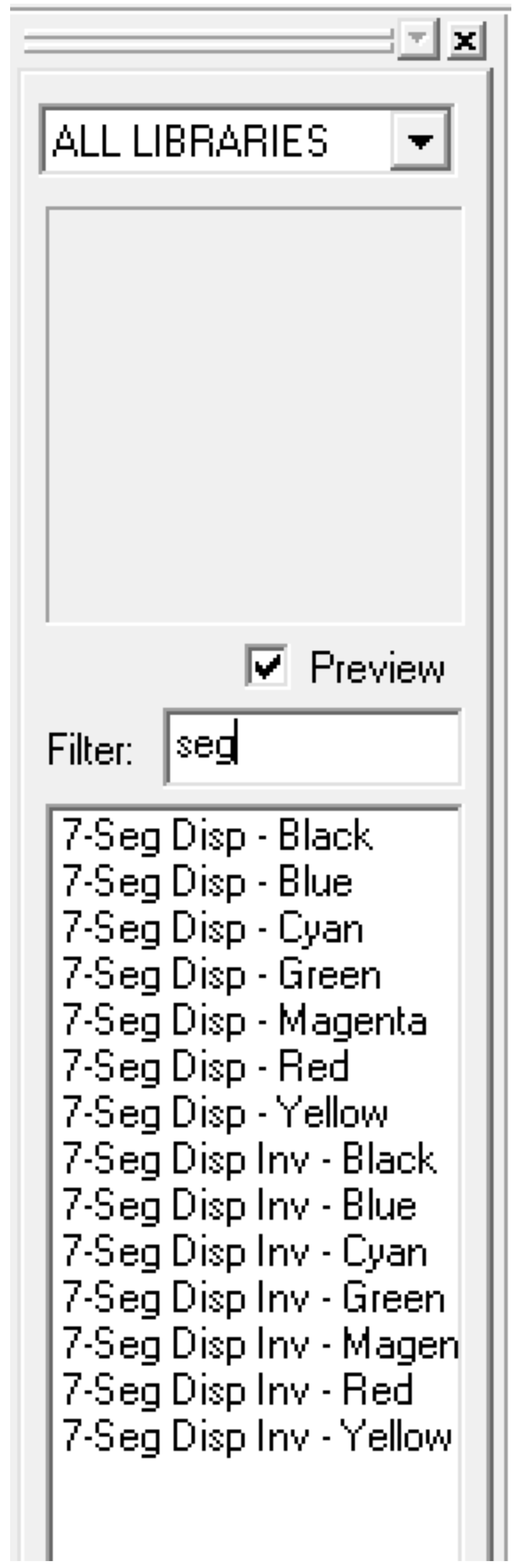
$$g = \boxed{\overline{A}C\overline{D}} + \boxed{\overline{A}\,\overline{B}C} + \boxed{\overline{A}B\overline{C}} + \boxed{A\overline{B}\,\overline{C}}$$

Example

P1
Example 3b

BCD to 7 Segment Display

- 'BCD-to-seven-segment *decoder*'
  - '*decodes*' binary code for decimal digit
  - usually decoders mean something different, which we’ll discuss later


ALL LIBRARIES

☑ Preview

Filter: seg

7-Seg Disp - Black
7-Seg Disp - Blue
7-Seg Disp - Cyan
7-Seg Disp - Green
7-Seg Disp - Magenta
7-Seg Disp - Red
7-Seg Disp - Yellow
7-Seg Disp Inv - Black
7-Seg Disp Inv - Blue
7-Seg Disp Inv - Cyan
7-Seg Disp Inv - Green
7-Seg Disp Inv - Magen
7-Seg Disp Inv - Red
7-Seg Disp Inv - Yellow

Example

P2
Example 3c

4-bit Equity Comparator

Ah, not again

Summary

P2
Example 3c

4-bit Equity Comparator

- Compare two numbers
- Equal or not

Example

P2
Example 3c

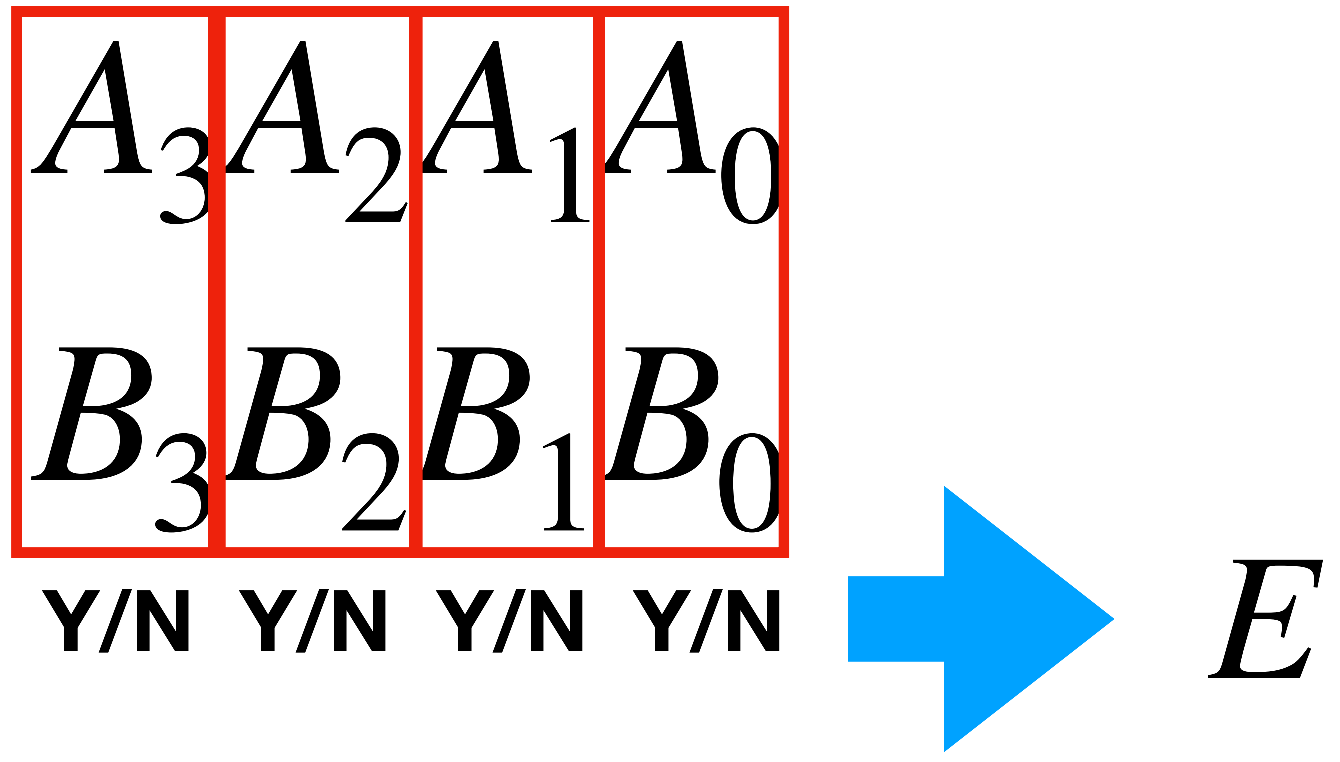
1. Specification

- Input: $A_3A_2A_1A_0$, $B_3B_2B_1B_0$
- Output: $E = 1$ for equal, 0 for not

Example

P2
Example 3c

2. Formulation



- 2 things are happening
  - We need to compare pairs of binary values and see if they are equal
  - We need to combine the comparison of all these different bits to make up E

Example

P2
Example 3c

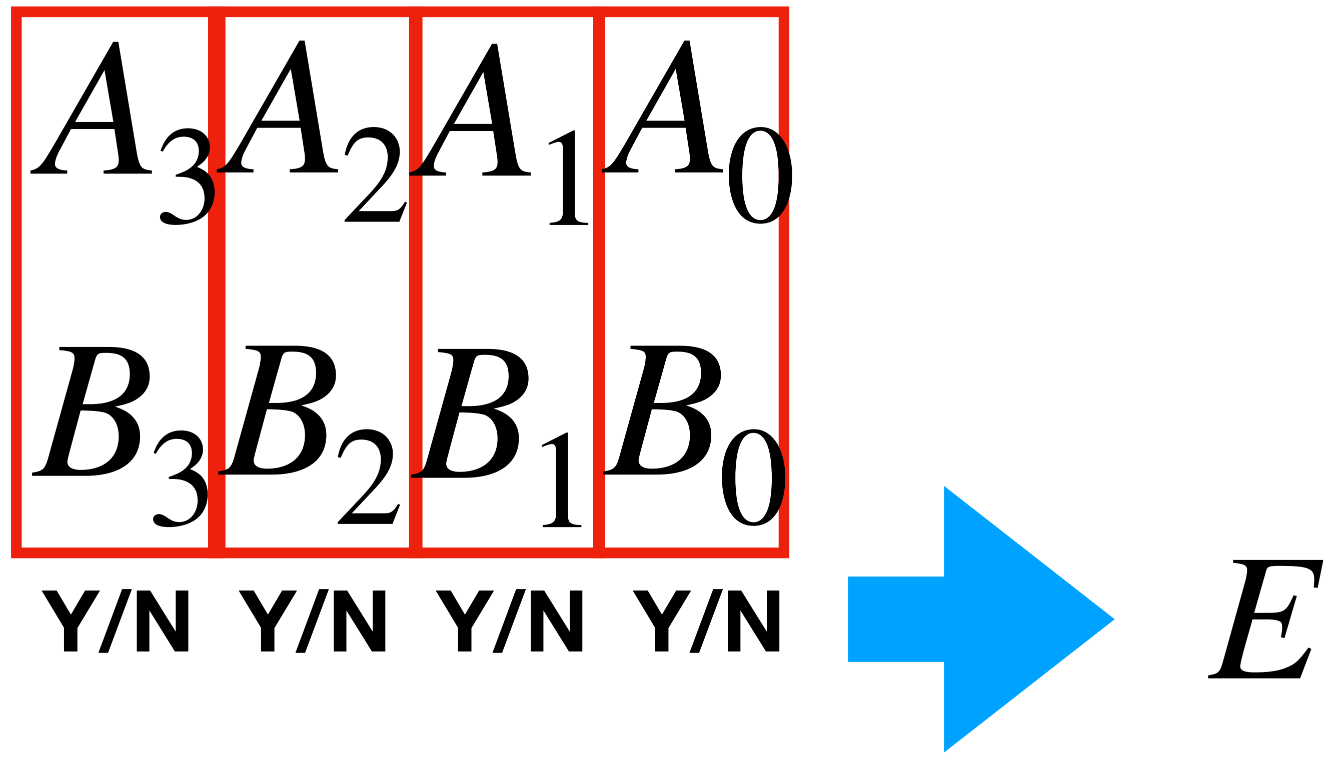
Hierarchical Design

- "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

Concept

P2
Example 3c

3. Optimisation

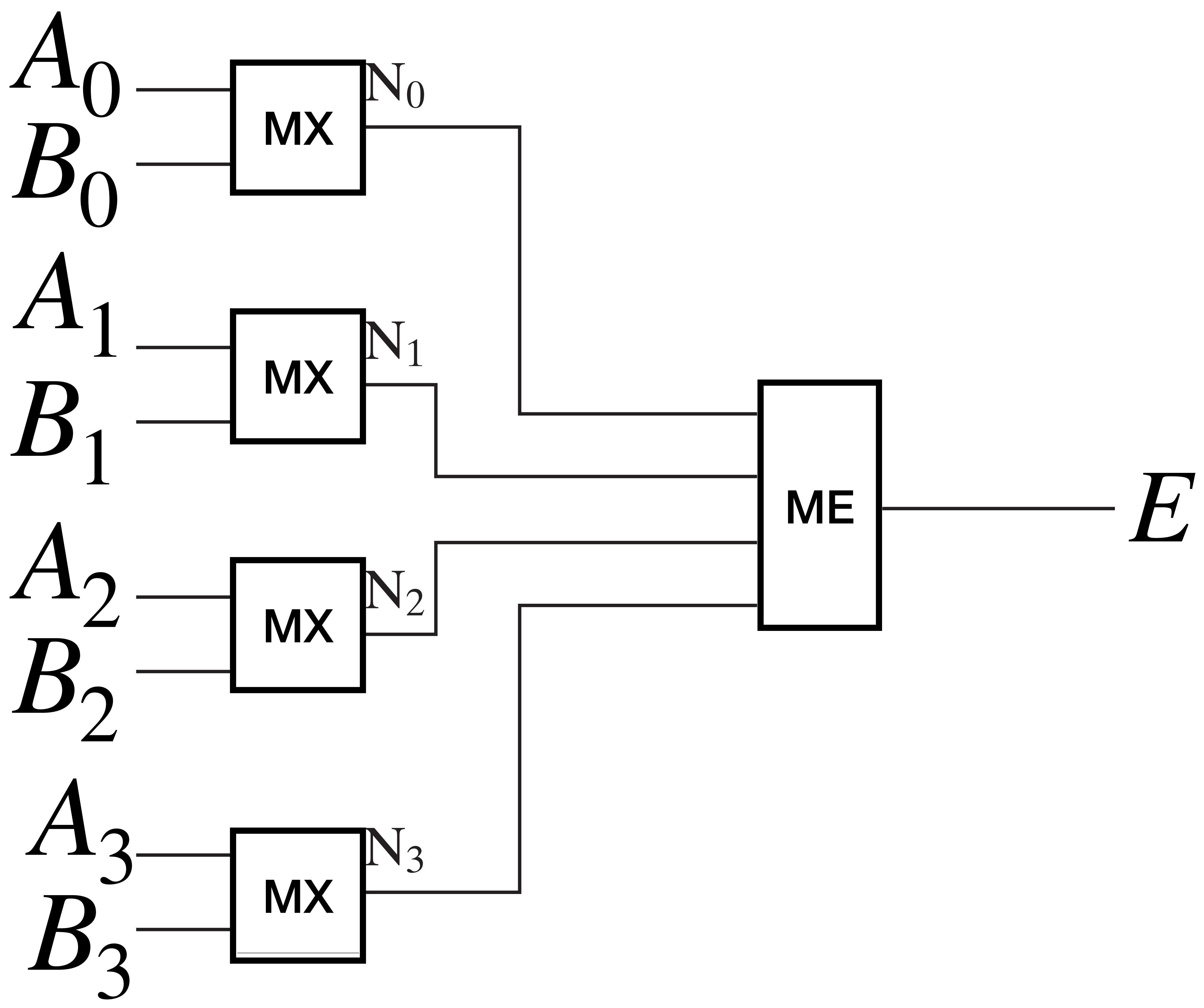


- 2 things are happening
  - We need to compare pairs of binary values and see if they are equal $\boxed{\text{MX}}$
  - We need to combine the comparison of all these different bits to make up E $\boxed{\text{ME}}$

Example

P2
Example 3c

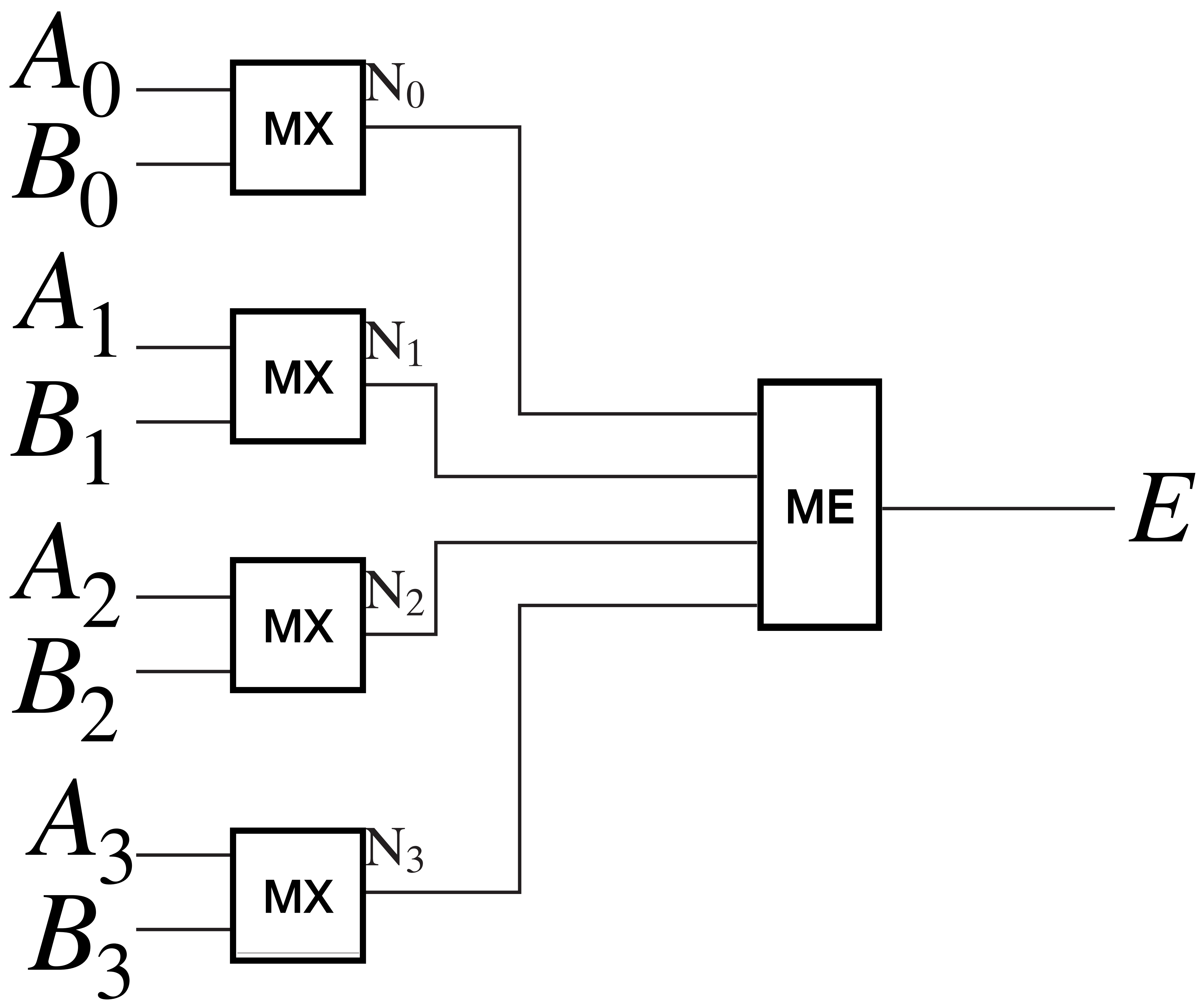
3. Optimisation



Example

P2
Example 3c

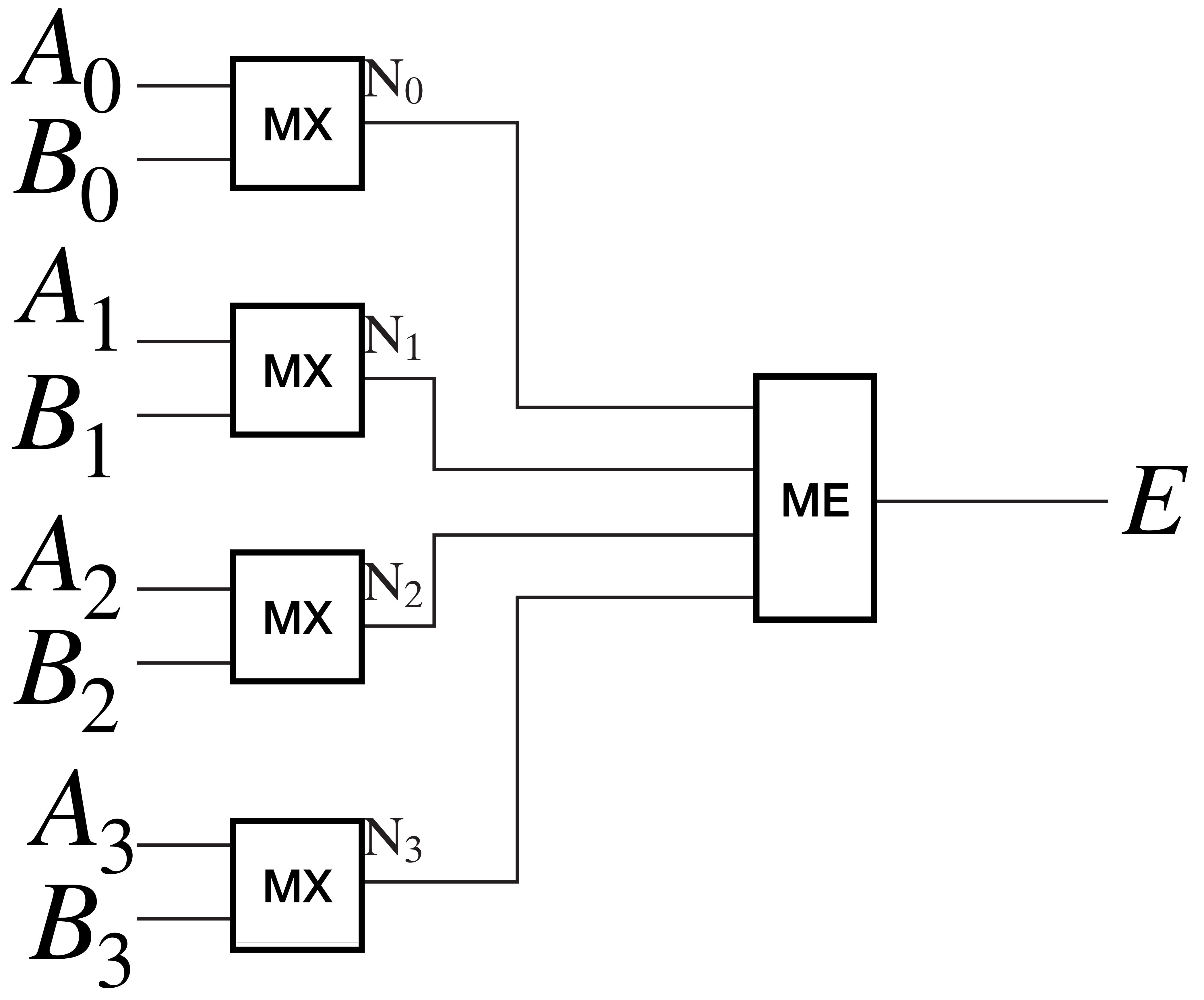
3. Optimisation

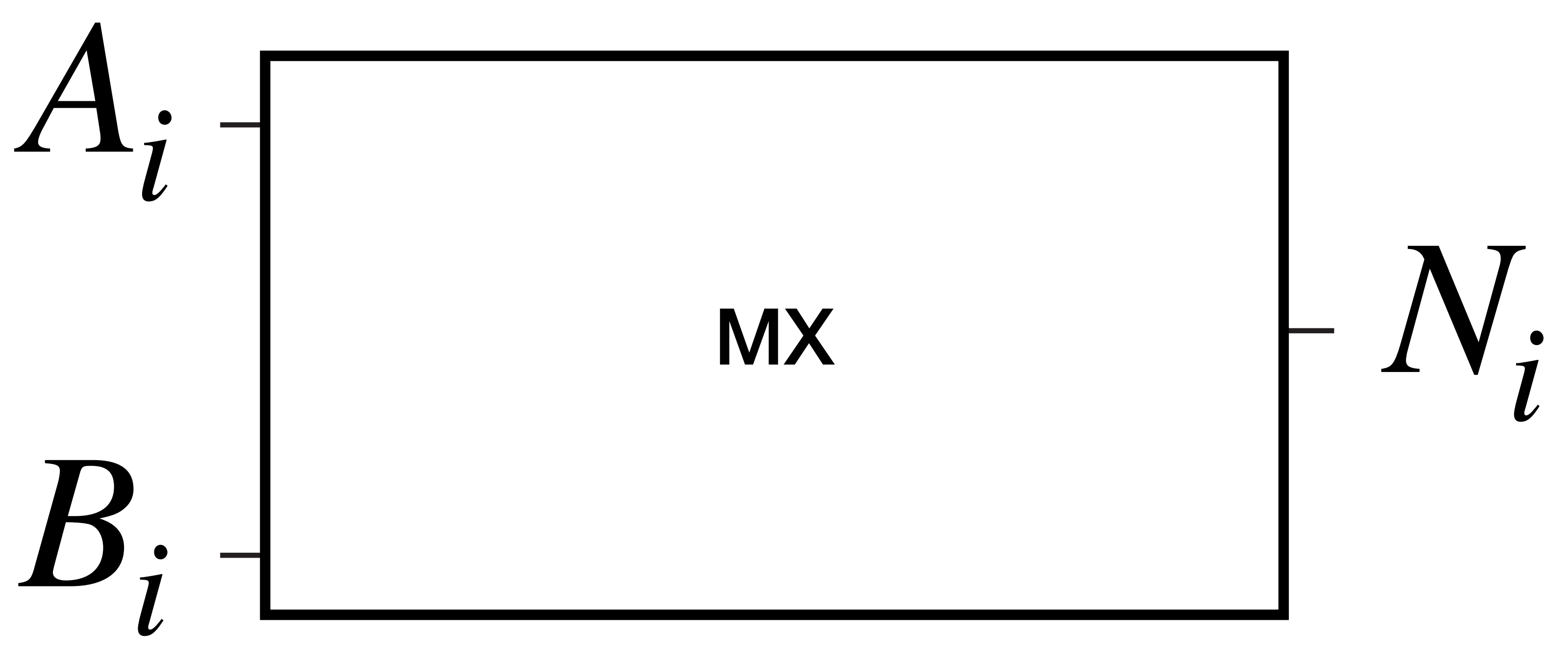


Example

P2
Example 3c

3. Optimisation





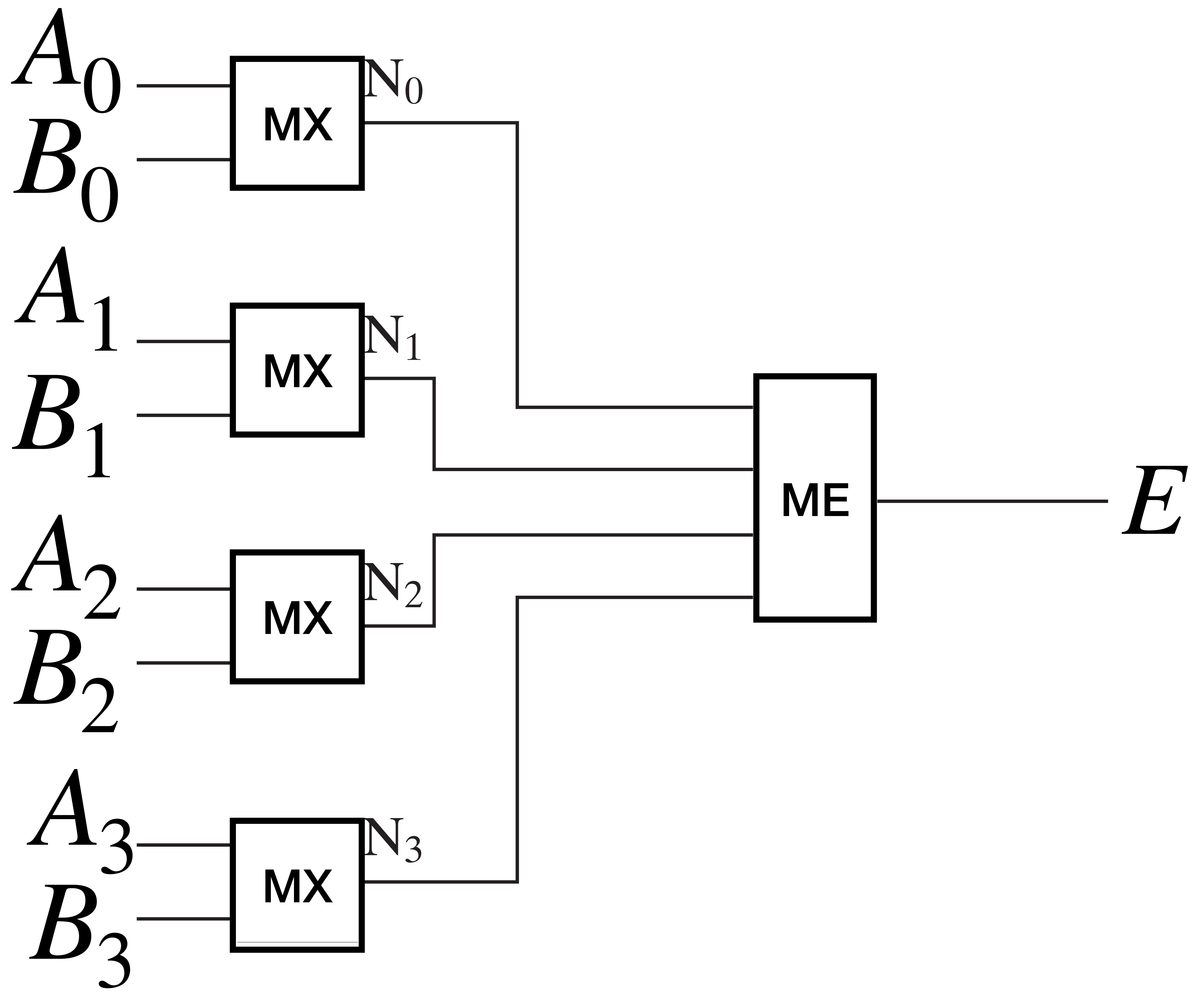
$MX(A_i, B_i) = N_i = 0$ if $A_i = B_i$

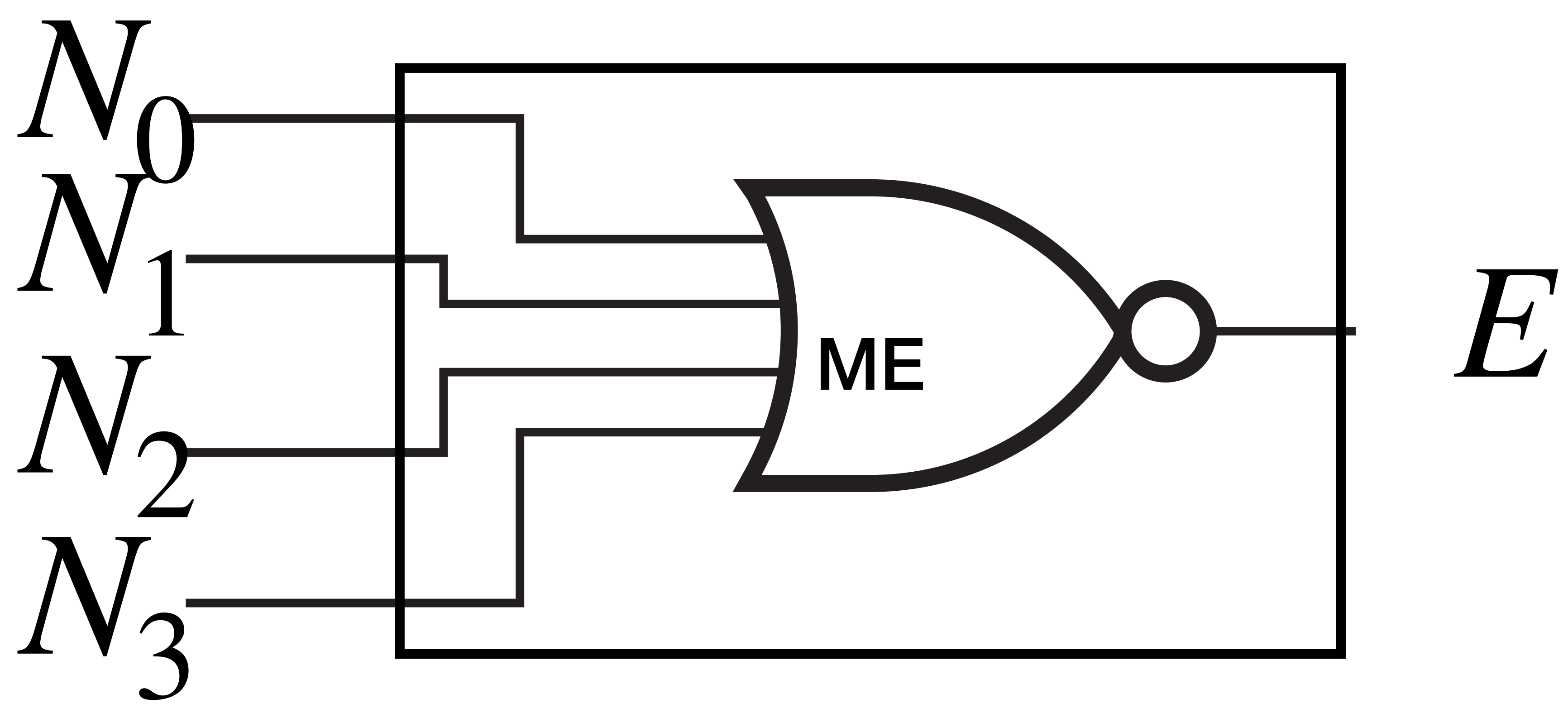
$MX(A_i, B_i) = N_i = \overline{A_i}B_i + A_i\overline{B_i}$

Example

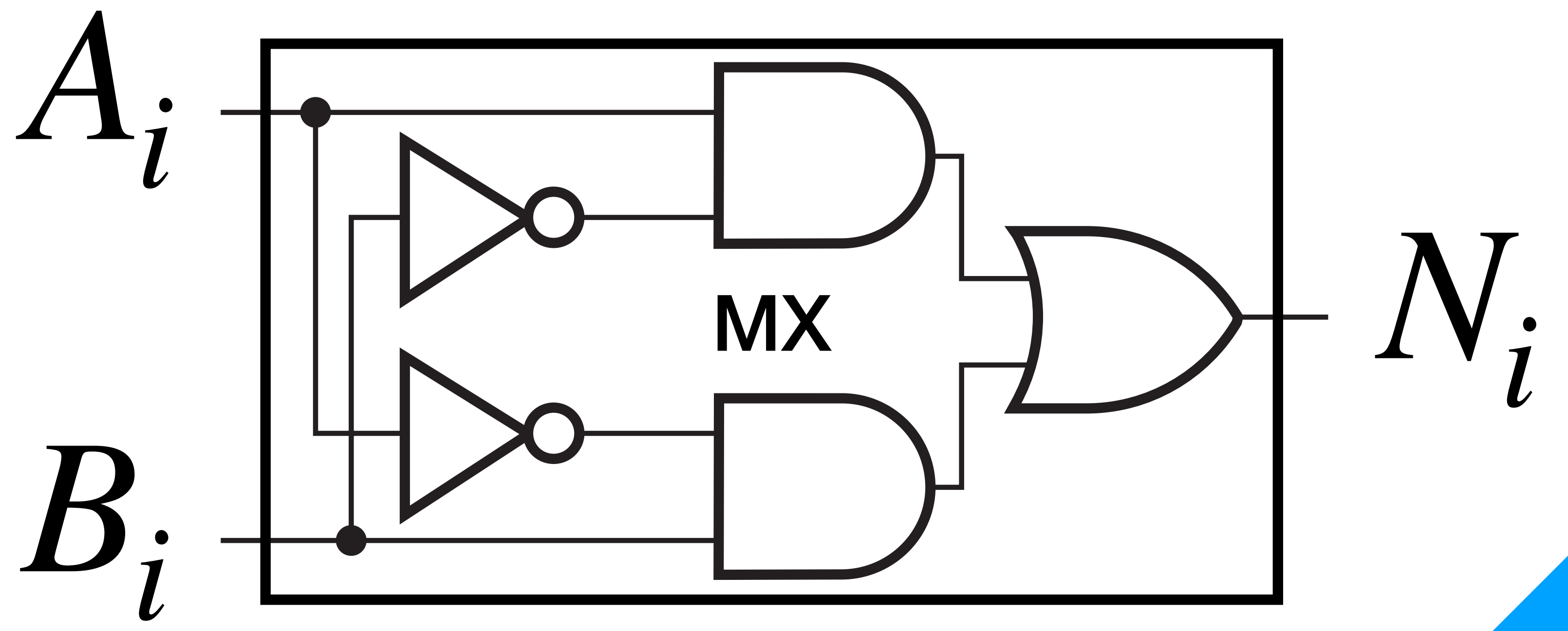
P2
Example 3c

3. Optimisation





$$ME(N_0, N_1, N_2, N_3) = E = \overline{N_0 + N_1 + N_2 + N_3}$$



$$MX(A_i, B_i) = N_i = \overline{A_i}B_i + A_i\overline{B_i}$$

Example

Technology Mapping

Mostly Old Tech, no new iPhones

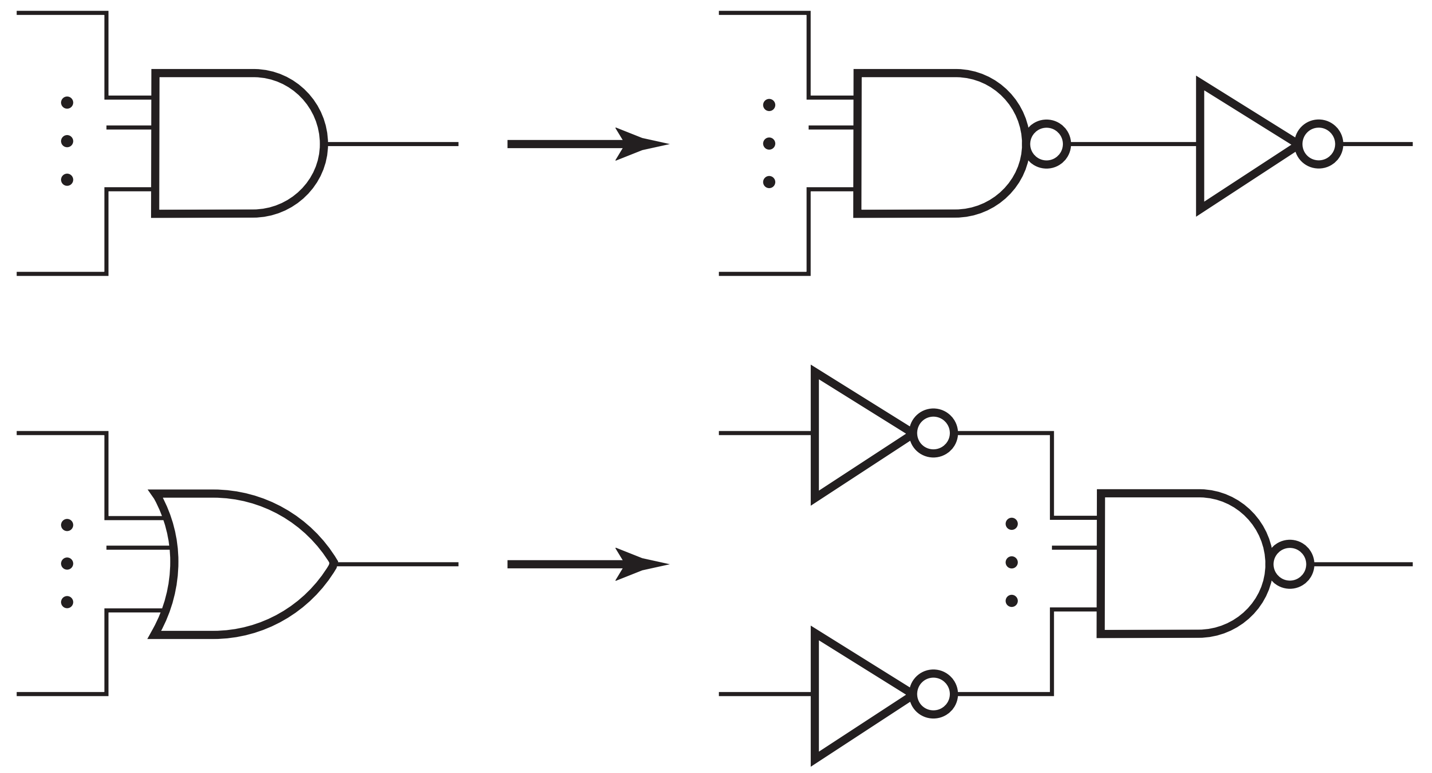
Technology Mapping

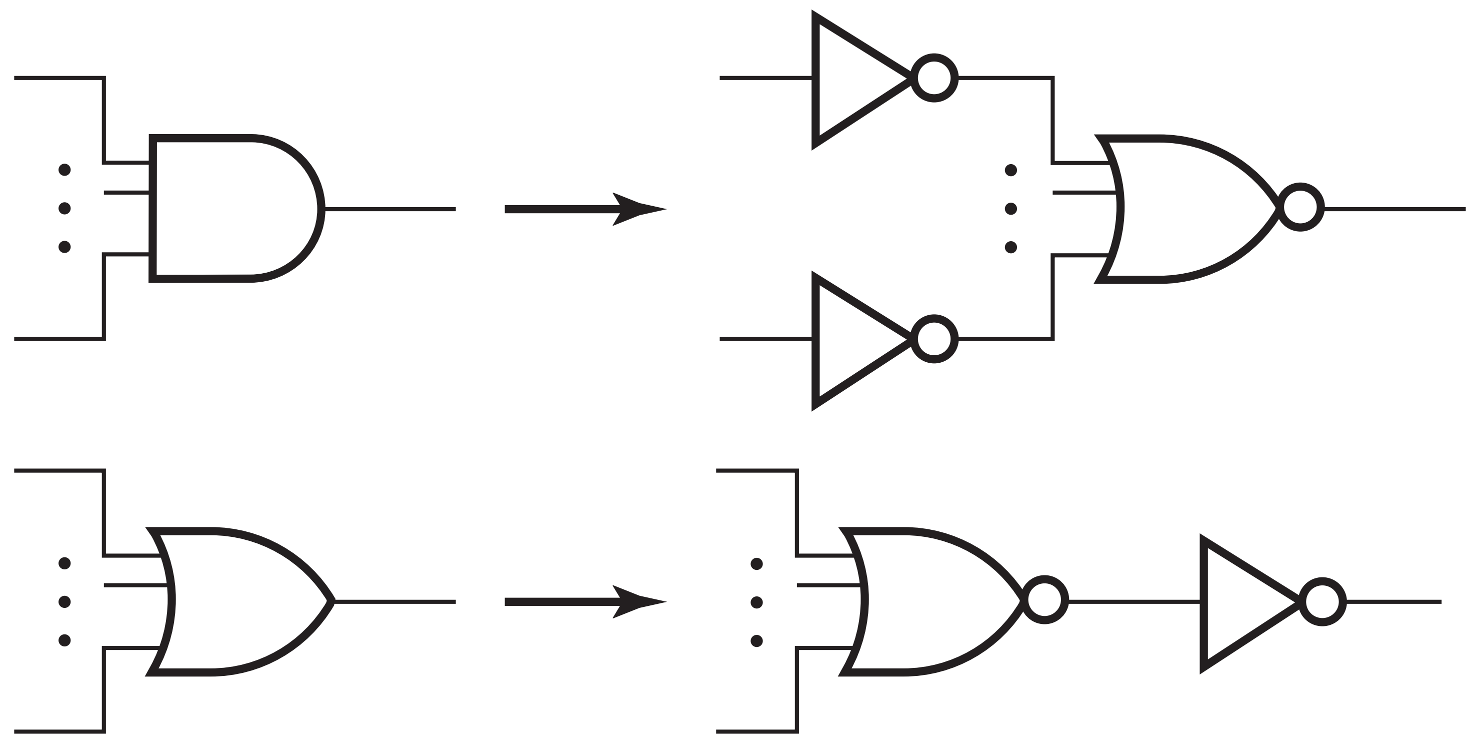
- Technology
  - Available physical components
  - Programmable implementation technology e.g. Field-Programmable Gate Array (FPGA)
    - VHDL; Verilog Language

P3
Tech. Mapping

Technology Mapping

- NAND == AND + Inverter;
  - AND == NAND + Inverter;
- NOR == OR + Inverter;
  - OR == NOR + Inverter;
- DeMorgen's Rule

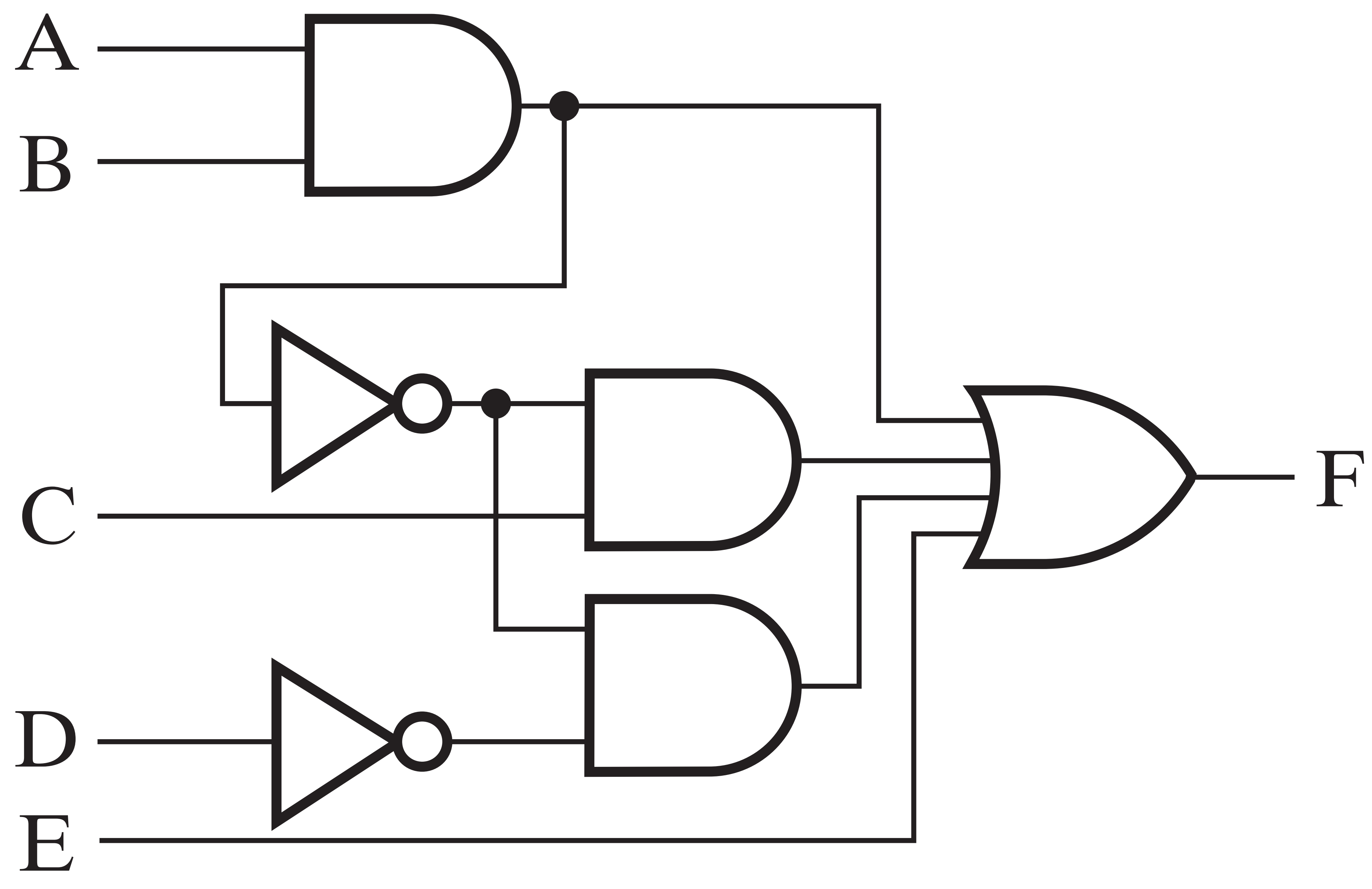

(a) Mapping to NAND gates


(b) Mapping to NOR gates

Concept

Technology Mapping

- Implement the circuit diagram with NAND gates and Inverters only



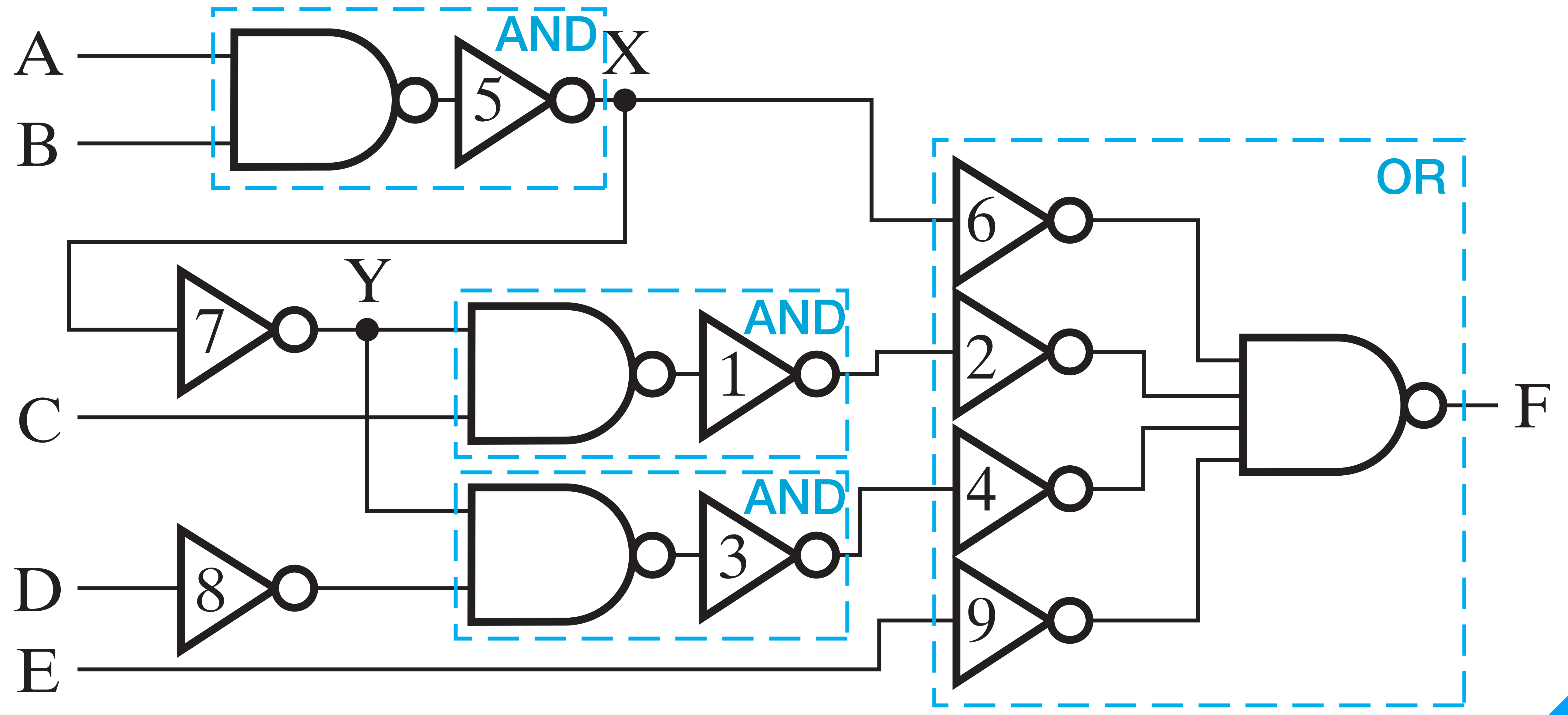
Example

P3
Tech. Mapping

Technology Mapping

- Implement the circuit diagram with NAND gates and Inverters only

1. Replacement

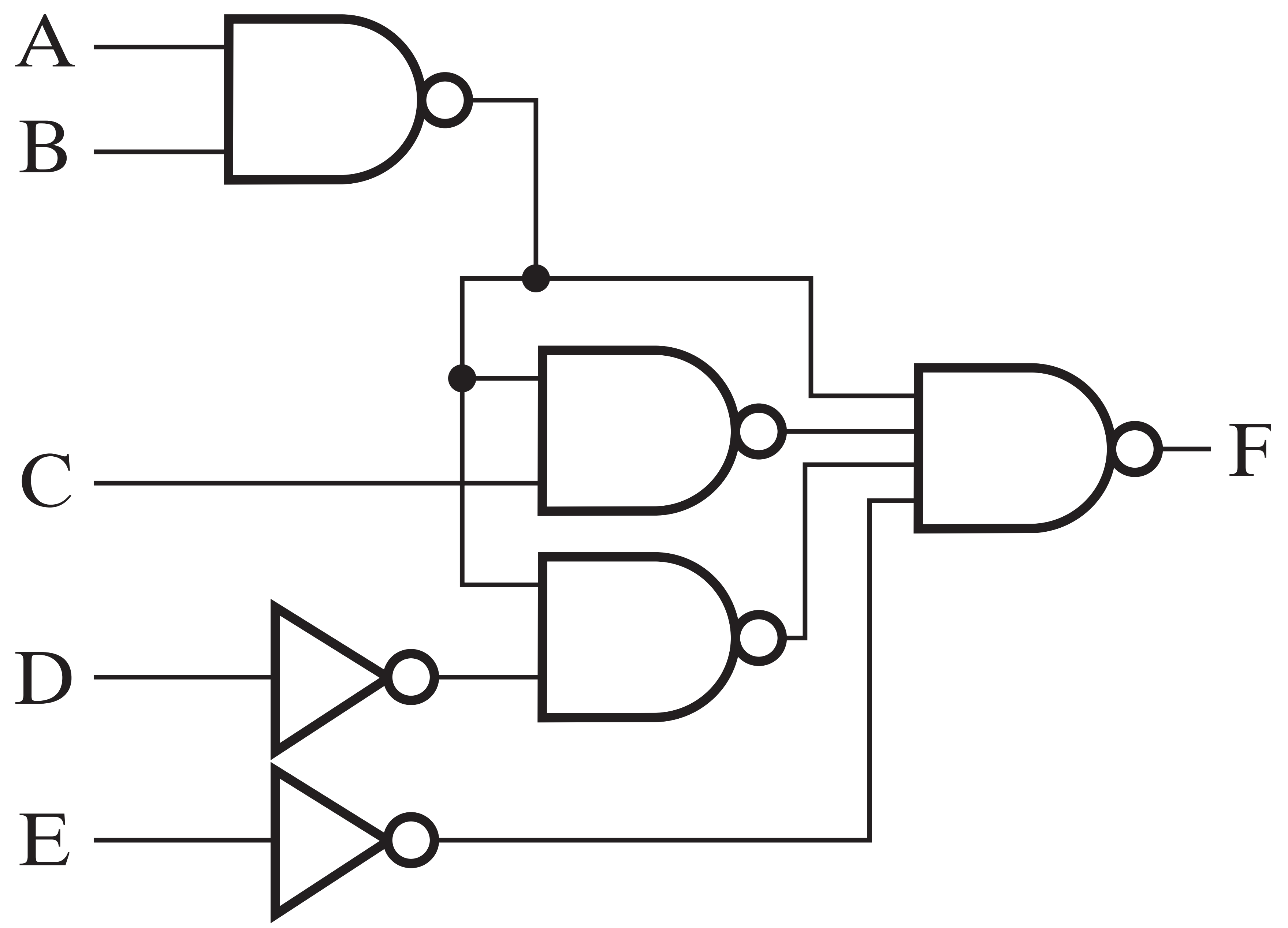


Example

Technology Mapping

- Implement the circuit diagram with NAND gates and Inverters only

1. Replacement
2. Simplification



Verification

Manual vs Auto

Summary

Verification

- Manual
  - Use the truth table, row by row
- Auto
  - Use computer simulation to go through the truth table

Concept