B38DB: Digital Design and Programming
Combinatorial Logic Design – Decoders, Multiplexers

Mustafa Suphi Erden

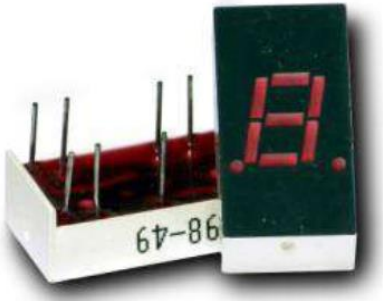
Heriot-Watt University
School of Engineering & Physical Sciences
Electrical, Electronic and Computer Engineering
Room: EM 2.01
Phone: 0131-4514159
E-mail: m.s.erden@hw.ac.uk

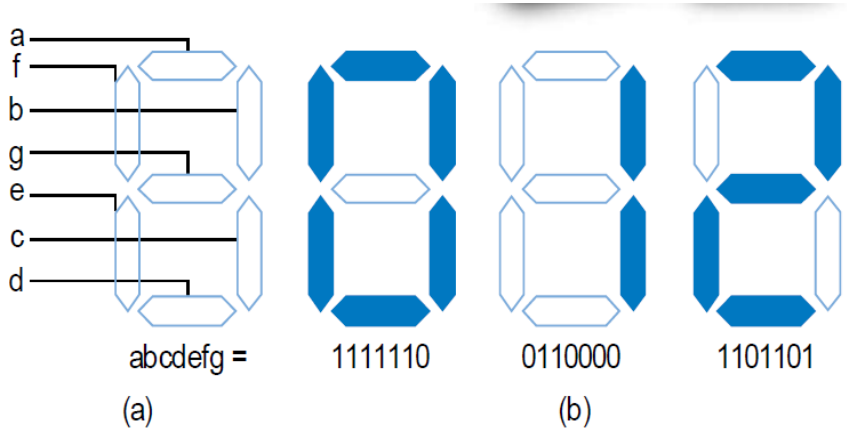
Reminder

Multiple-Output Example: BCD to 7-Segment Converter

TABLE 2-4 4-bit binary number to seven-segment display truth table

w	x	y	z	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0





a = w'x'y'z' + w'x'yz' + w'x'yz + w'xy'z + w'xyz' + w'xyz + wx'y'z' + wx'y'z

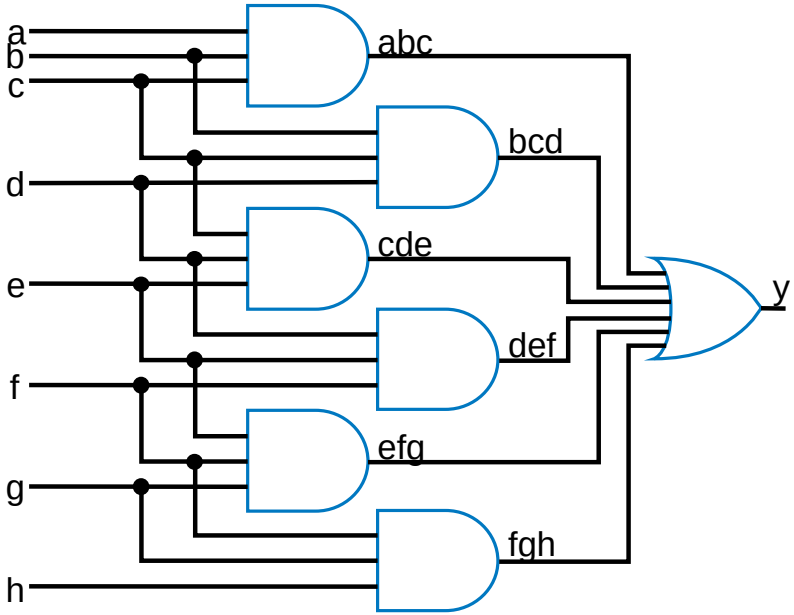
b = w'x'y'z' + w'x'y'z + w'x'yz' + w'x'yz + w'xy'z' + w'xyz + wx'y'z' + wx'y'z

Combinational Logic Design Process

	Step	Description
Step 1	**Capture** the function	Create a truth table or equations, ***whichever is most natural for the given problem***, to describe the desired behavior of the combinational logic.
Step 2	**Convert** to equations	This step is only necessary if you captured the function using a truth table instead of equations. Create an equation for each output by **OR**ing all the minterms for that output. Simplify the equations if desired.
Step 3	**Implement** as a gate-based circuit	For each output, create a circuit corresponding to the output's equation. (Sharing gates among multiple outputs is OK optionally.)

Example: Three 1s Detector

- Problem: Detect three consecutive 1s in 8-bit input: **abcdefgh**
  - 000**111**01 → 1 10101011 → 0
    11110000 → 1
  - **Step 1: Capture** the function
    - Truth table or equation?
      - Truth table too big: 2^8=256 rows
      - Equation: create terms for each possible case of three consecutive 1s
    - **y = abc + bcd + cde + def + efg + fgh**
  - **Step 2: Convert** to equation – already done
  - **Step 3: Implement** as a gate-based circuit



Boolean Functions

- F_1= a **AND** b

a	b	F_1
0	0	0
0	1	0
1	0	0
1	1	1

- F_2= a′ **OR** b

a	b	F_2
0	0	1
0	1	1
1	0	0
1	1	1

- How many possible functions of 2 variables?

Number of Possible Boolean Functions

- How many possible functions of 2 variables?
  - 2^2 rows in truth table, 2 choices for each
  - $2^{(2^2)} = 2^4 = 16$ possible functions

a	b	F	
0	0	0 or 1	2 choices
0	1	0 or 1	2 choices
1	0	0 or 1	2 choices
1	1	0 or 1	2 choices

$2^4 = 16$ possible functions

a	b	f0	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
		0	a AND b		a		b	a XOR b	a OR b	a NOR b	a XNOR b	b'		a'		a NAND b	1

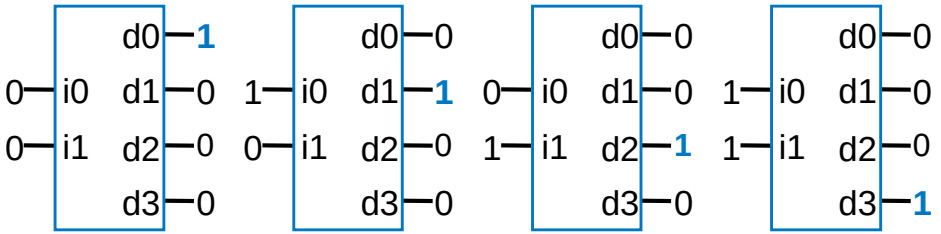
- N variables
  - 2^N rows
  - $2^{(2^N)}$ possible functions

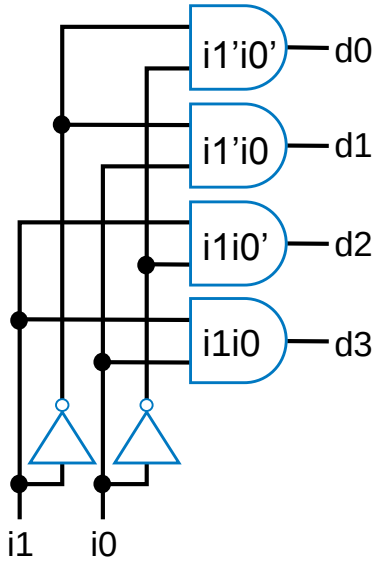
Decoders

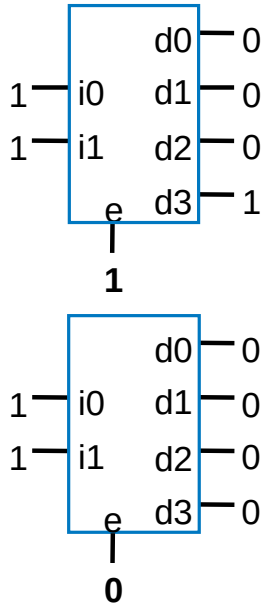
- **Decoder**:

  Relates an input binary number to one of the outputs

- 2-input decoder:
  - Four possible input binary numbers
  - Four outputs, one for each possible input binary number
- Decoder with enable e
  - Outputs all 0 if e=0
  - Regular behavior if e=1
- n-input decoder: 2^n outputs

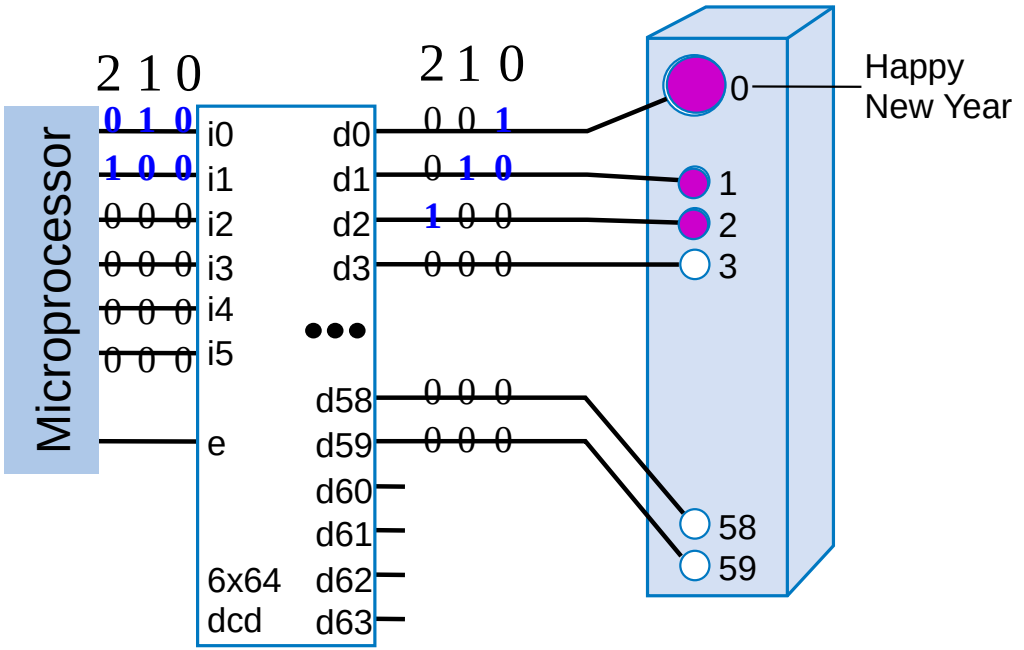






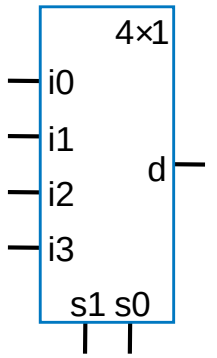
Decoder Example

- New Year's Eve Countdown Display
  - Microprocessor counts from 59 down to 0 in binary on 6-bit output
  - Want to illuminate one of 60 lights for each binary number
  - Use 6x64 decoder

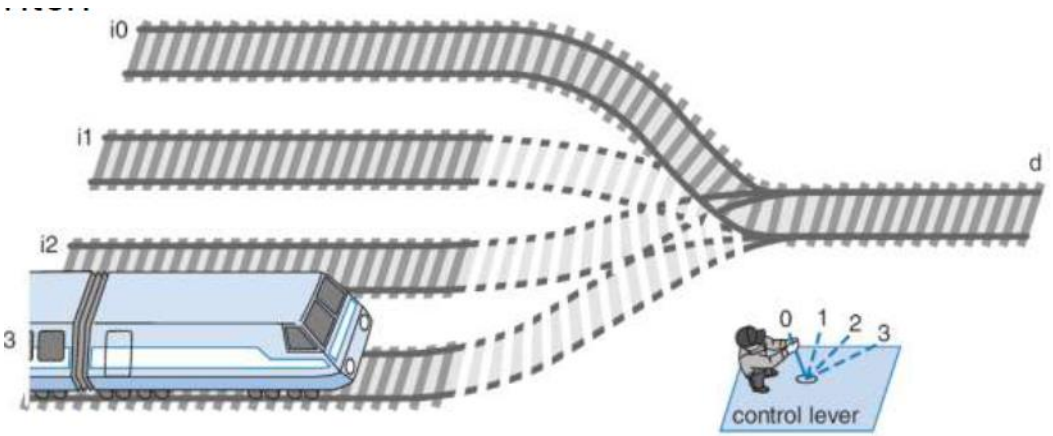


Multiplexor (Mux)

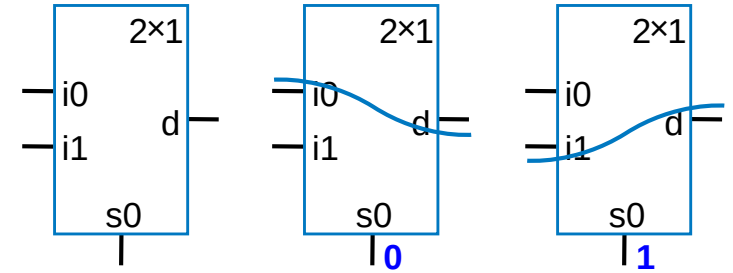
- Mux:
  - Routes one of its N data inputs to its one output, based on binary value of select inputs
    - 4 input mux → needs 2 select inputs to indicate which input to route through
    - 8 input mux → 3 select inputs
    - N inputs → $\log_2(N)$ selects
  - Like a railyard switch



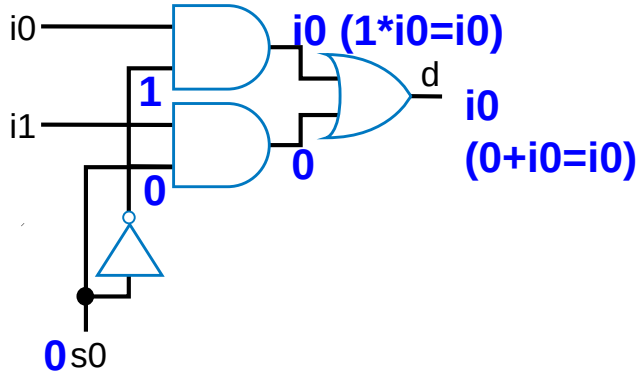
4x1 mux

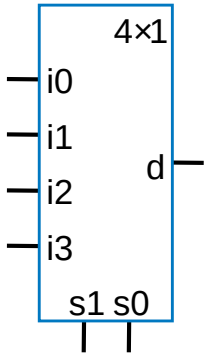


Mux Internal Design

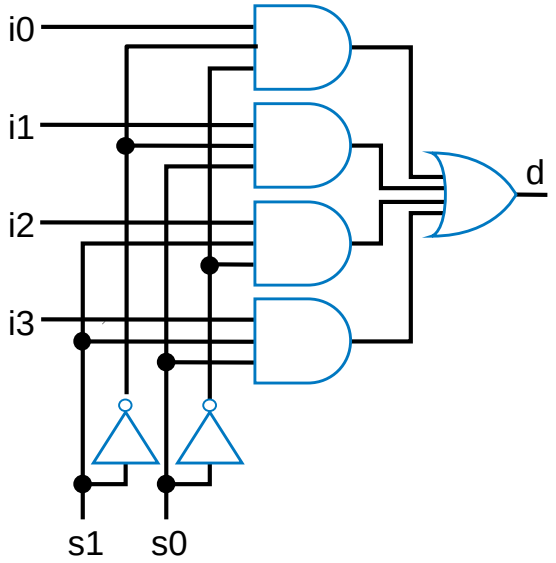


2x1 mux



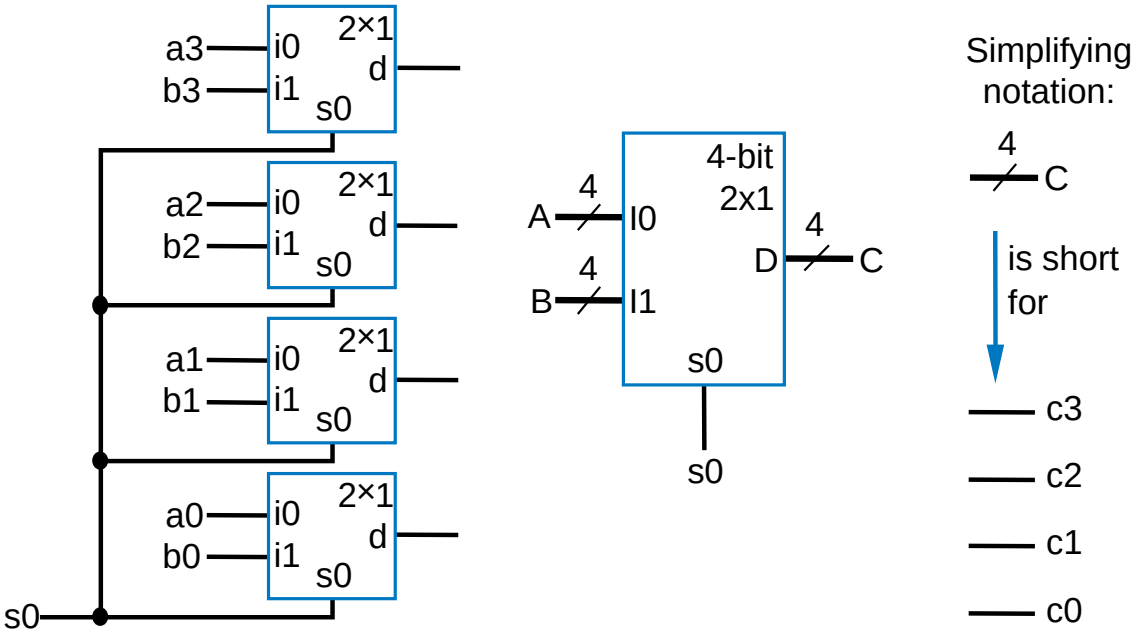


4x1 mux

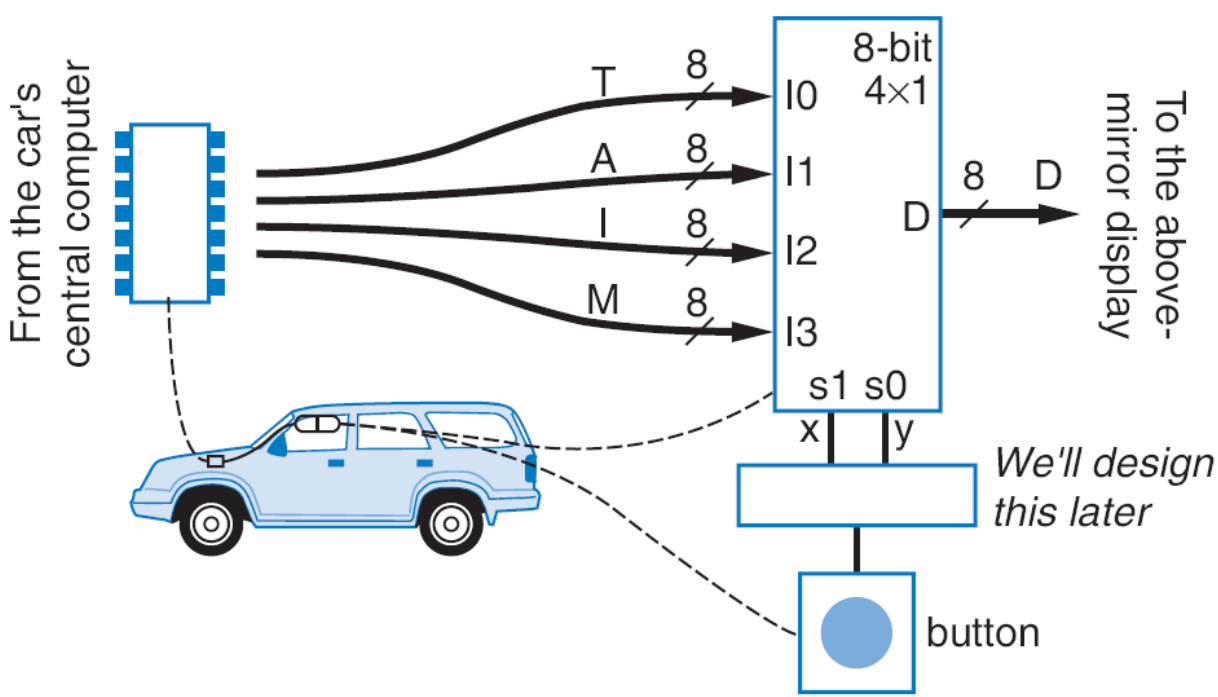


Muxes Combined Together – N-bit Mux

- Ex: Two 4-bit inputs, A (a3 a2 a1 a0), and B (b3 b2 b1 b0)
  - 4-bit 2x1 mux (just four 2x1 muxes sharing a select line) can select between A or B



N-bit Mux Example



- Four possible display items – each is 8-bits length
  - Temperature (T),
  - Average miles-per-gallon (A),
  - Instantaneous mpg (I), and
  - Miles remaining (M)