

Lecture 4 — Combinational Logic

Table of contents

Today's outline	1
Recap — binary adder	2
Binary subtractors	2
Combinational circuits in general	2
Multiplexers	2
Quiz: 2-to-1 multiplexer truth table	3
Quiz: choosing mux inputs to realise $X = \bar{S}_0 + S_1$	3
Decoders	4
Timing diagram representations	4
Quiz: timing diagram for an XOR gate	4
Reminders	5

See combinational-logic-blocks¹ for the reference definitions/tables of the adder-subtractor, multiplexer and decoder introduced here, and timing-diagrams² for the timing-diagram representation and real-gate timing parameters.

Today's outline

- Admin
- Binary subtractors
- More combinational logic circuits
 - Multiplexers
 - Decoders
 - Timing diagram representations

¹courses/csse2010/combinational-logic-blocks.pdf

²courses/csse2010/timing-diagrams.pdf

Recap — binary adder

Picking up from 2026-08-04-binary-arithmetic³: full adders can be **cascaded** to make a multi-bit binary adder. For 4 bits, four full adders take $A_3B_3 \dots A_0B_0$ and produce $S_3 \dots S_0$ plus C_4 , with each stage's carry-out C_{i+1} wired to the next stage's carry-in. For addition the initial carry-in C_0 is 0. This arrangement is a **ripple-carry adder**.

Binary subtractors

$A - B$ is implemented as $A + (-B)$, with $-B$ the two's complement of B (flip the bits, add 1). The two pieces needed on top of a plain adder are a **controlled inverter** (an XOR gate per bit, with the mode select M as the second input) and the extra +1, which is supplied via the adder's carry-in. Full derivation and the adder-subtractor circuit are in combinational-logic-blocks⁴.

Two slides here are posed as questions and left blank on the handout:

- “How can we use a gate to flip a bit — but only sometimes?” ($Z = \bar{B}$ when $M = 1$, $Z = B$ when $M = 0$.) Derived: a 2-input XOR, $Z = B \oplus M$.
- “What should carry-in be?” on the adder-subtractor slide. Derived: $C_0 = M$ — the same mode-select line that drives the XOR gates, so subtract mode automatically contributes the +1 that two's complement needs.

Combinational circuits in general

A combinational circuit is a combination of logic gates with n inputs and m outputs; each output is a function of the n inputs, the **output depends on current inputs only**, and it can be written as a truth table with n input columns, m output columns and 2^n rows. See combinational-logic-blocks⁵.

Multiplexers

2^n data inputs, 1 output, n select inputs that steer one data input to the output. The lecture covered the 4-to-1 mux (logic symbol, function table, and the AND-OR sum-of-products implementation with inverted select literals) and the 2-to-1 mux. Tables and circuit descriptions: combinational-logic-blocks⁶. Off-the-shelf part: the 74HCT157 quad 2-input mux, see device-pinouts⁷.

³courses/csse2010/2026-08-04-binary-arithmetic.pdf

⁴courses/csse2010/combinational-logic-blocks.pdf

⁵courses/csse2010/combinational-logic-blocks.pdf

⁶courses/csse2010/combinational-logic-blocks.pdf

⁷courses/csse2010/device-pinouts.pdf

Quiz: 2-to-1 multiplexer truth table

A 2-to-1 multiplexer has data inputs D_0 and D_1 , control input S_0 , and output F , with function table $F = D_0$ when $S_0 = 0$ and $F = D_1$ when $S_0 = 1$. Which of three candidate truth tables (columns S_0, D_0, D_1, F) goes with this circuit?

Reasoning from the function table, row by row in the order $S_0 D_0 D_1 = 000, 001, 010, 011, 100, 101, 110, 111$:

- The first four rows have $S_0 = 0$, so F must copy D_0 and ignore D_1 : $F = 0, 0, 1, 1$.
- The last four rows have $S_0 = 1$, so F must copy D_1 and ignore D_0 : $F = 0, 1, 0, 1$.

$$F = 0, 0, 1, 1, 0, 1, 0, 1$$

That is **option 2**. Option 1 has F following D_1 when $S_0 = 0$ and D_0 when $S_0 = 1$ — the two data inputs swapped. Option 3 is $F = 0, 0, 0, 0, 1, 1, 1, 1$, i.e. $F = S_0$: it ignores the data inputs completely.

Equivalently: $F = D_0 \bar{S}_0 + D_1 S_0$.

Quiz: choosing mux inputs to realise $X = \bar{S}_0 + S_1$

A 4-to-1 multiplexer has A on data input 0, B on 1, C on 2, D on 3, selects $S_1 S_0$, and output X . What must A, B, C, D be so that

$$X = \bar{S}_0 + S_1$$

Options: (1) $A=0, B=1, C=0, D=0$; (2) $A=0, B=1, C=1, D=1$; (3) $A=1, B=0, C=1, D=1$; (4) $A=1, B=1, C=0, D=1$.

Reasoning: the mux steers data input i to the output when $S_1 S_0$ equals i in binary, so each data input must simply be **the value the desired function takes for that select combination**. Evaluate $\bar{S}_0 + S_1$ for all four:

S_1	S_0	$\bar{S}_0$	$X = \bar{S}_0 + S_1$	Selected input
0	0	1	1	A (input 0)
0	1	0	0	B (input 1)
1	0	1	1	C (input 2)
1	1	0	1	D (input 3)

So $A = 1, B = 0, C = 1, D = 1$ — **option 3**.

This is the general trick: tying a mux's data inputs to constants turns it into a lookup table for any function of the select variables.

Decoders

A decoder converts an n -bit input to a logic 1 on exactly one of 2^n outputs. The lecture used the 3-to-8 decoder (inputs A, B, C ; outputs $D_0 \dots D_7$) with its truth table and the eight-AND-gate implementation. See combinational-logic-blocks⁸.

Timing diagram representations

The lecture closed by adding the **timing diagram** as another logic representation — like a truth table but graphical — using an inverter as the worked example, then the convention for drawing 2-input waveforms so all input combinations are covered, then the reality that gates aren't perfect (propagation delay, fall time, rise time). All of this is in timing-diagrams⁹.

Quiz: timing diagram for an XOR gate

Given input waveforms where A is low for the first half of the diagram then high for the second half, and B toggles at twice that rate (low, high, low, high) — so the four intervals are $AB = 00, 01, 10, 11$ — which of three candidate output waveforms represents an **XOR** gate?

Reasoning from the XOR truth table ($X = 1$ iff exactly one input is 1):

Interval	A	B	$A \oplus B$
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

So the output must be **low, high, high, low** — a single pulse spanning the middle two intervals. That is **option 2** — the only candidate with a pulse that goes high and comes back low again. Option 1 has a single rising edge and then stays high to the end, so it can never be XOR. Option 3 is the exact complement of option 2 (high, low, low, high), i.e. XNOR.

⁸courses/csse2010/combinational-logic-blocks.pdf

⁹courses/csse2010/timing-diagrams.pdf

Reminders

- **Lab 4 preparation task** — next week (week 3). It will be available on Blackboard under *Learning Resources*; it involves drawing some circuit schematic diagrams (see [circuit-schematics¹⁰](#)), and you must bring it to Lab 4 (P2 Mon-Tue sessions in week 3). Complete it before Lab 4.
- Supporting material for the labs is now on Blackboard.
- Attempt the posted exercises and quiz (not assessed).
- Any doubts: use course consultation, ask after the lecture, or use the Ed forum.

¹⁰[courses/csse2010/circuit-schematics.pdf](#)