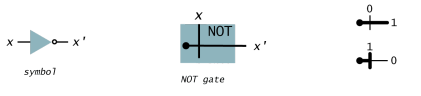
6.3 Sequential Circuits (plus a few Combinational)

Logic Gates: Fundamental Building Blocks

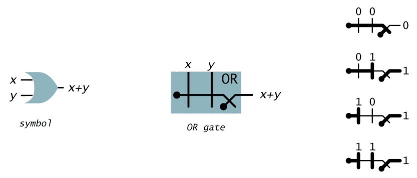
NOT = x'

x	NOT
0	1
1	0



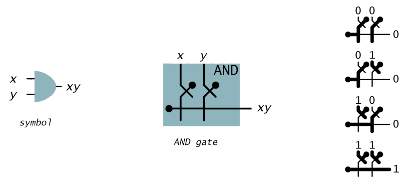
OR = x+y

x y	OR
0 0	0
0 1	1
1 0	1
1 1	1



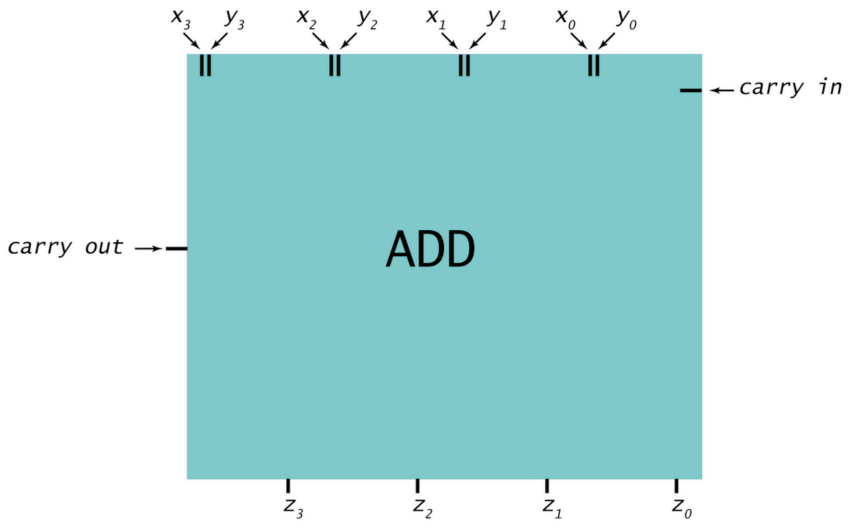
AND = xy

x y	AND
0 0	0
0 1	0
1 0	0
1 1	1

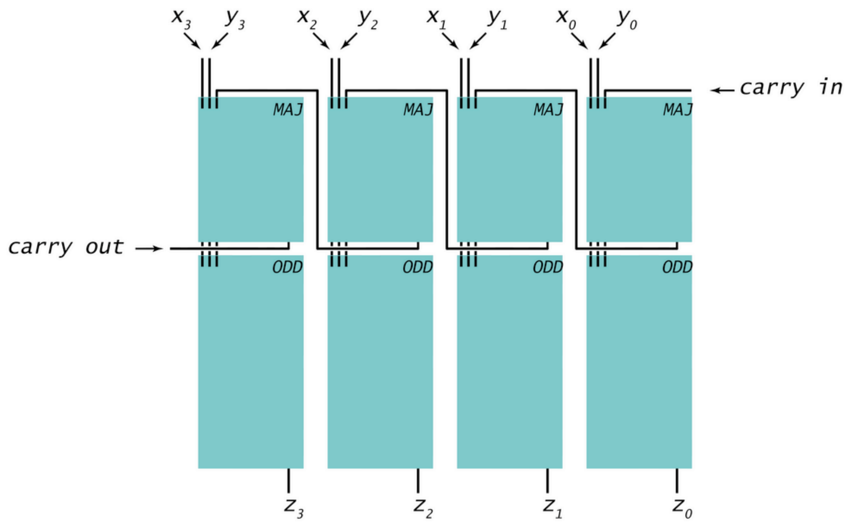


implementations with switches

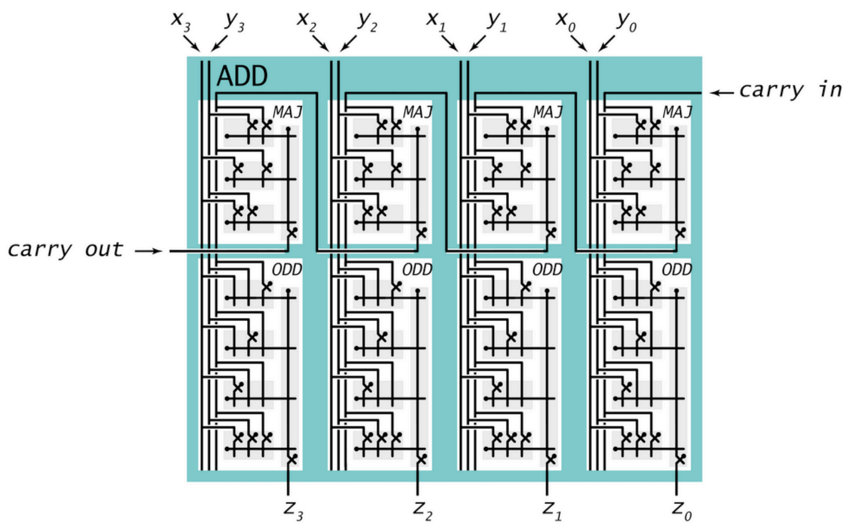
Adder: Interface



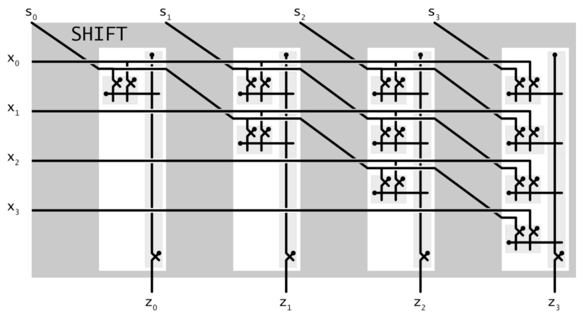
Adder: Component Level View



Adder: Switch Level View



(Right) Shifter

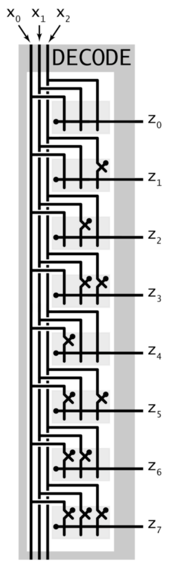


4-bit Shifter

Decoder

Decoder. [n-bit]

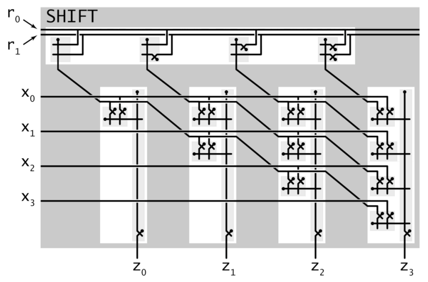
- n address inputs, 2^n data outputs.
- Addressed output bit is 1; others are 0.



3-bit Decoder

2-Bit Decoder Controlling 4-Bit Shifter

Ex. Put in a binary amount to shift.



Right-shifter with decoder

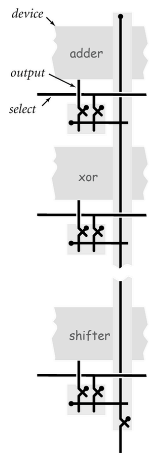
Arithmetic Logic Unit

Arithmetic logic unit (ALU). Computes all operations in parallel.
- Add and subtract.
- Xor.
- And.
- Shift left or right.

Q. How to select desired answer?

1 Hot OR

1 hot OR.
- All devices compute their answer; we pick one.
- Exactly one select line is on.
- Implies exactly one output line is relevant.

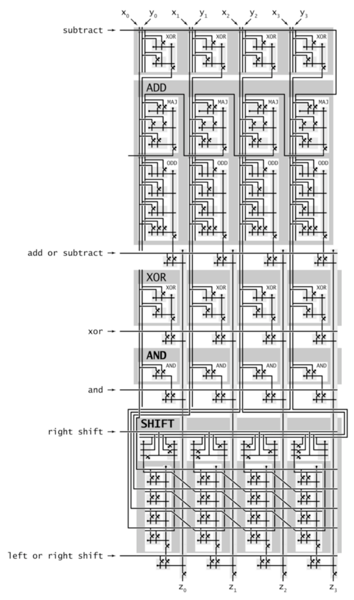


Output select with one-hot OR

ALU

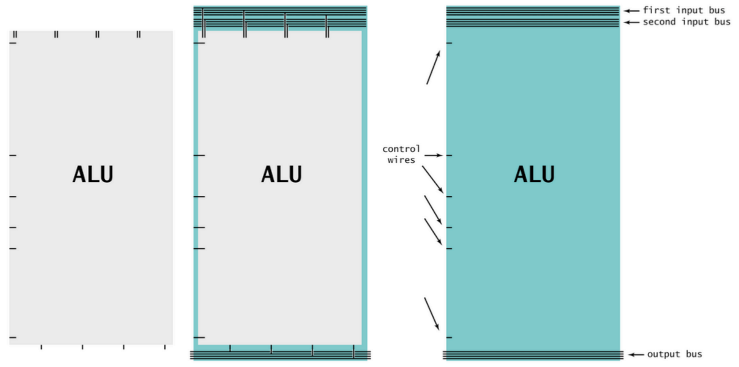
Arithmetic logic unit.
- Add and subtract.
- Xor.
- And.
- Shift left or right.

Arithmetic logic unit.
- Computes all operations in parallel.
- Uses 1-hot OR to pick each bit answer.

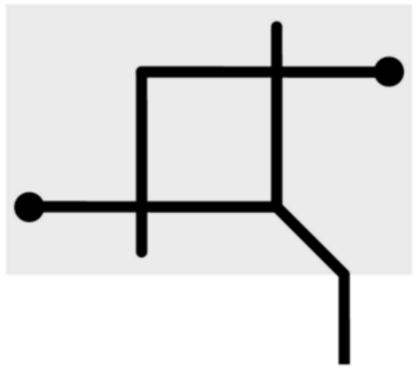


Device Interface Using Buses

Device. Processes a word at a time. ← 16-bit words for TOY memory
Input bus. Wires on top.
Output bus. Wires on bottom.
Control. Individual wires on side.

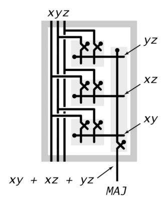


6.3 Sequential Circuits

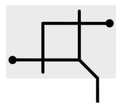


Sequential vs. Combinational Circuits

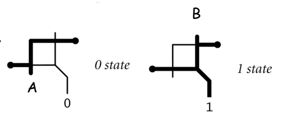
Combinational circuits.
- Output determined solely by inputs.
- Can draw with no loops.
- Ex: majority, adder, ALU.



Sequential circuits.
- Output determined by inputs and previous outputs.
- Ex: memory, program counter, CPU.

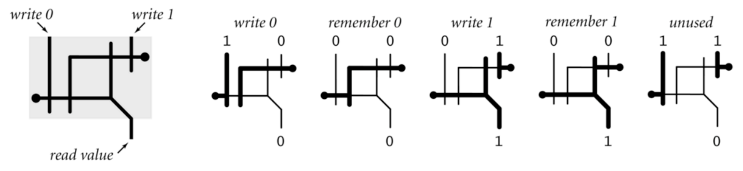


Ex. Simplest feedback loop.
- Two relays A and B, both connected to power, each blocked by the other.
- State determined by whichever switches first.
- Stable.



Flip-Flop

Flip-flop.
- A way to control the feedback loop.
- Abstraction that "remembers" one bit.
- Basic building block for memory and registers.



Caveat. Need to deal with switching delay.

Memory Overview

Computers and TOY have several memory components.
- Program counter.
- Registers.
- Main memory.

Implementation. Use one flip-flop for each bit of memory.

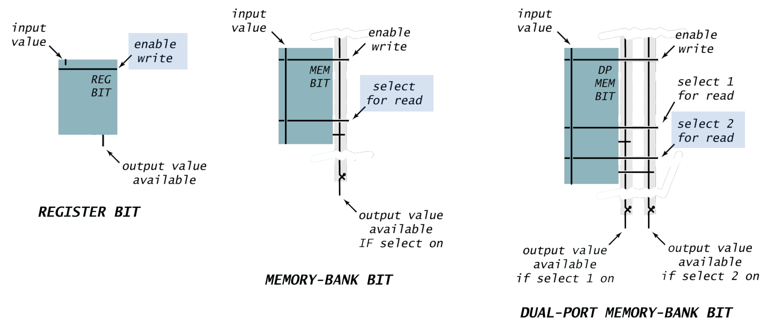
Access. Memory components have different access mechanisms.

Organization. Need mechanism to manipulate groups of related bits.

TOY has 16 bit words,
8 bit memory addresses, and
4 bit register names.

Memory Bit: Interface

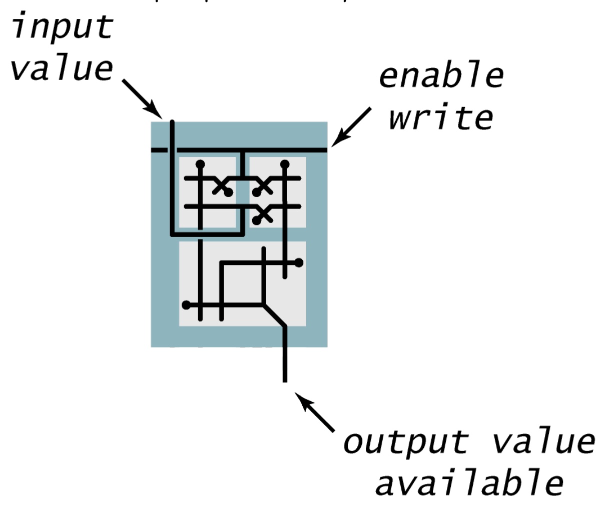
Memory bit. Extend a flip-flop to allow easy access to values.



(TOY PC, IR) (TOY main memory) (TOY registers)

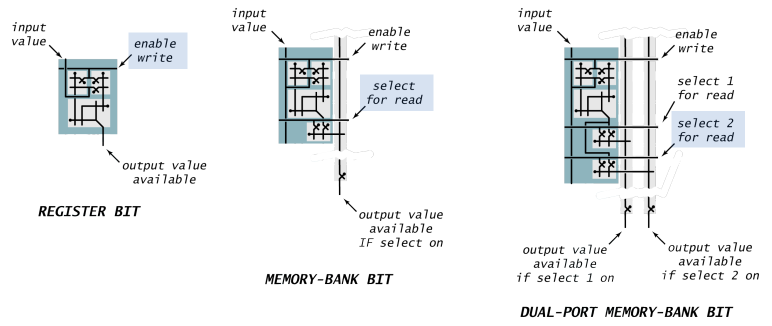
Memory Bit: Switch Level Implementation

Memory bit. Extend a flip-flop to allow easy access to values.



Memory Bit: Switch Level Implementation

Memory bit. Extend a flip-flop to allow easy access to values.



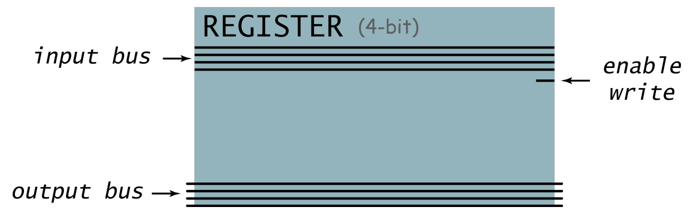
[TOY PC, IR] [TOY main memory] [TOY registers]

Processor Register

Processor register. ← don't confuse with TOY register

- Stores k bits.
- Register contents always available on output bus.
- If enable write is asserted, k input bits get copied into register.

Ex 1. TOY program counter (PC) holds 8-bit address.
Ex 2. TOY instruction register (IR) holds 16-bit current instruction.

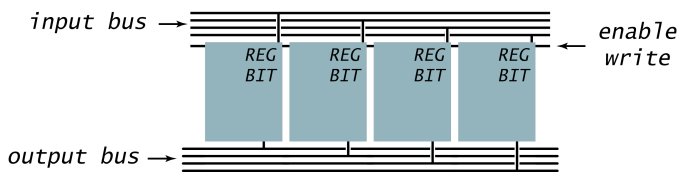


Processor Register

Processor register. ← don't confuse with TOY register

- Stores k bits.
- Register contents always available on output bus.
- If enable write is asserted, k input bits get copied into register.

Ex 1. TOY program counter (PC) holds 8-bit address.
Ex 2. TOY instruction register (IR) holds 16-bit current instruction.

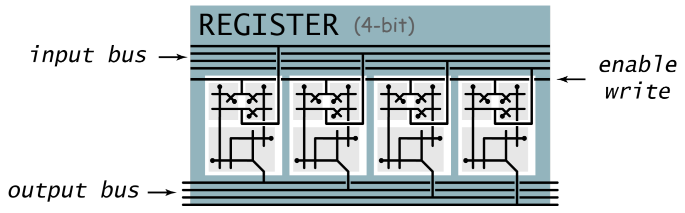


Processor Register

Processor register. ← don't confuse with TOY register

- Stores k bits.
- Register contents always available on output bus.
- If enable write is asserted, k input bits get copied into register.

Ex 1. TOY program counter (PC) holds 8-bit address.
Ex 2. TOY instruction register (IR) holds 16-bit current instruction.



Memory Bank

Memory bank.

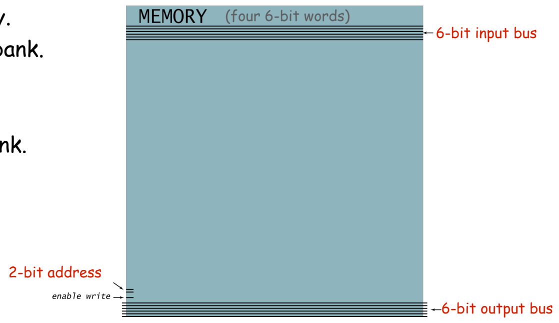
- Bank of n registers; each stores k bits.
- Read and write information to *one* of n registers.
- Address inputs specify which one. ← $\log_2 n$ address bits needed
- Addressed bits always appear on output.
- If write enabled, k input bits are copied into addressed register.

Ex 1. TOY main memory.

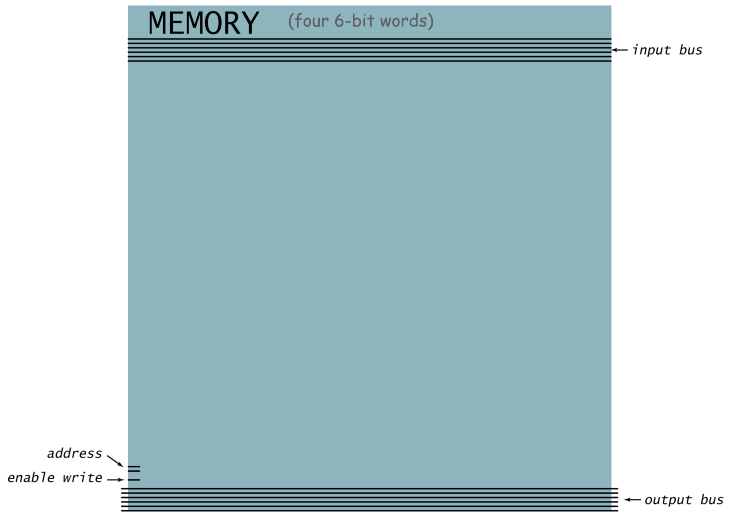
- 256-by-16 memory bank.

Ex 2. TOY registers.

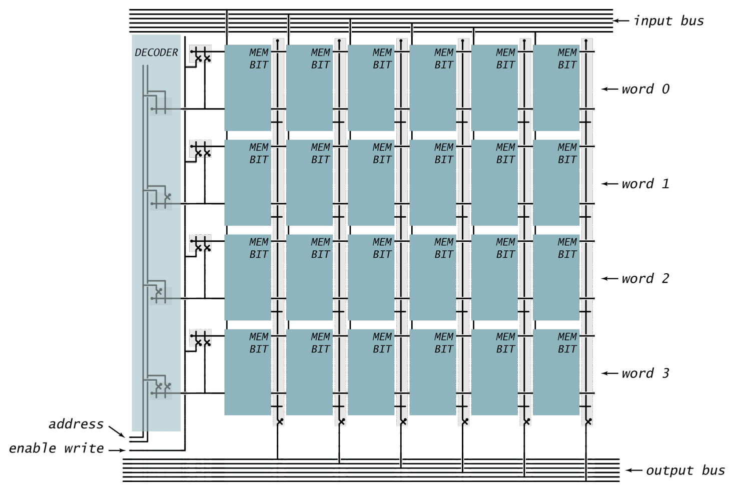
- 16-by-16 memory bank.
- Two output buses.



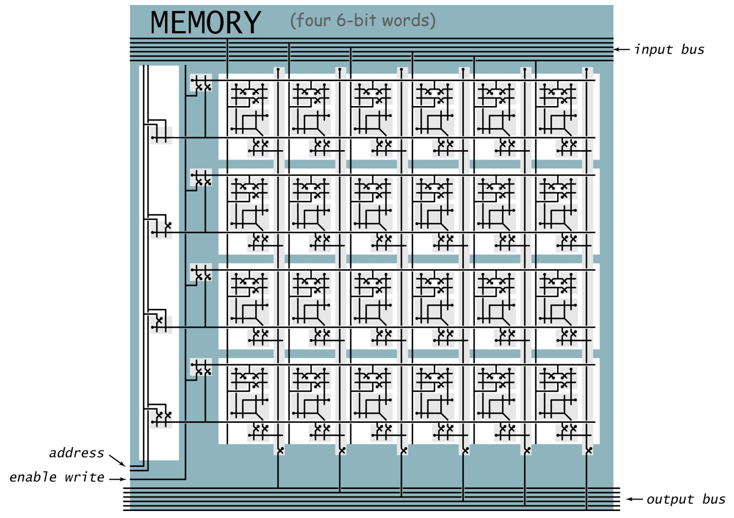
Memory: Interface



Memory: Component Level Implementation



Memory: Switch Level Implementation



Summary

Sequential circuits add "state" to digital hardware.

Flip-flop.	represents 1 bit
TOY word.	16 flip-flops
TOY registers.	16 words
TOY main memory.	256 words

Modern technologies for registers and main memory are different.

- Few registers, easily accessible, high cost per bit.
- Huge main memories, less accessible, low cost per bit.
- Drastic evolution of technology over time.

Next time. Build a complete TOY computer.