Basic Components

Components of a Computer

Combinational Structures/Circuits

- Always gives the same output for a given set of inputs
- Do not store any information
- Examples: adder, subtracter, multiplexer (mux), decoder

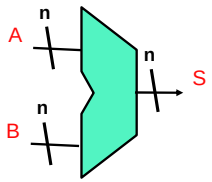
Sequential Structures/Circuits

- Combination of Combinational Structure and Memory Elements
- Store Information
- Example: memory, and state machine

Adder

Adder

- A and B are operands
- S is the result of the addition
- Write now we are not interested in how it works (CIT 595 goes in detail)

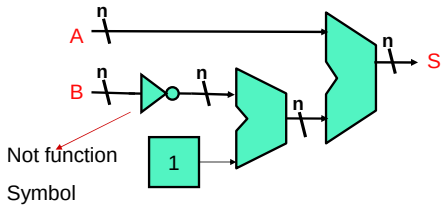


Black box view of the Adder

Subtracter

Build a subtracter from an adder

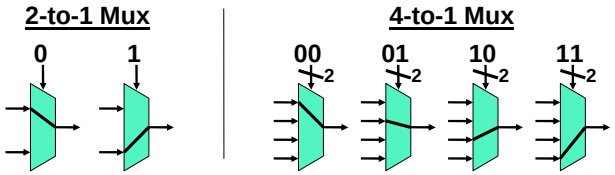
- Calculate A - B = A + -B
- Recall 2's complement!!
- Negate B
- Recall -B = NOT(B) + 1



Multiplexer (MUX)

Selector/Chooser of value (electrical voltages)

- Multi-way switch



In general

- N select bits chooses from 2^N inputs
- An incredibly useful building block

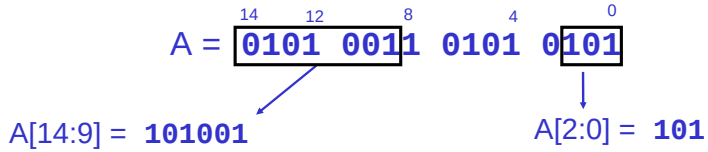
More on Representing Multi-bit Values

Number bits from right (0) to left (n-1)

- Just a convention -- could be left to right, but must be ***consistent***

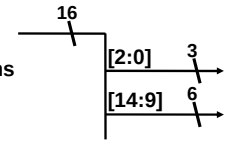
Use brackets to denote range:

- D[l:r] denotes bit l to bit r, from *left* to *right*



May also see A<14:9>

- Especially in hardware block diagrams



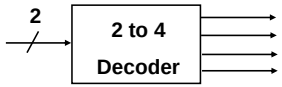
Decoder

Decodes information

- Undoing the encoding so that the original information can be retrieved

***n-bit* inputs, 2^n outputs**

- Exactly one output is 1 or ON or TRUE for each possible input pattern
- Example: 2 to 4 decoder



Overview of Sequential Structure/Circuit

Sequential Structure/Circuit

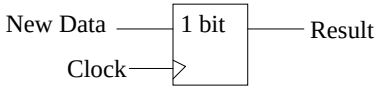
- Stores information – contains storage element
- Output depends on stored information (state) plus input
  - Given input might produce different outputs, depending on stored information
  - State = snapshot of all relevant elements of system at moment snapshot is taken
- Example:
  - state = Score board of basketball game (number of points, time remaining, possession)
  - input = which team scored the point
  - output = point increase for the team that just scored

Useful for building memory (ROM/RAM/Registers) and state machines (E.g. Vending Machine)

Storage Unit

Storage Unit a.k.a a flip-flop

- Stores 1 bit of Information
- Data is changed on an event (a.k.a clock*)
  - Do not want to change data abruptly
  - Event provides consitency

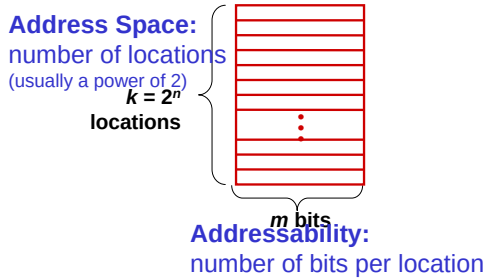


*clock is a pulse that generated using electronic circuitry
if clock = '1' then new date stored else stored value remains

Sequential: Memory

Memory (logical view)

- Holds information (e.g. phone directory contains phone numbers)
- a logical *k by m* array of stored bits



Memory Size

Terminology First

- Decimal system short form for large denominations
  - 1000 = 1K(thousand), 1,000,000 = 1M (million)
- Binary Systems sort notation for large denominations
  - 1 Byte = 2^3 = 8 bits
  - 1 Kilo Byte (KB) = 1024 bytes = 2^{10} Bytes = 2^{10} x 2^3 = 2^{13} = 8192 bits
  - 1 Mega Byte (MB) = 1024 KB = 2^{20} Bytes = 2^{20} x 2^3 = 2^{23} = 8388608 bits
  - 1Giga Byte (GB) = 1024 MB = 2^{30} Bytes = 2^{30} x 2^3 = 2^{33} = 8589934592 bits

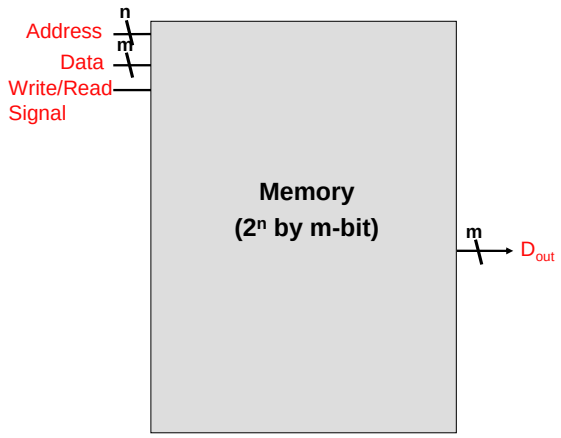
Example: 16 MB memory = 2^4 x 2^{20} = 2^{24} bytes

- 2^{24} = 16777216 (~ 16 million) unique locations, each holds a 8-bit value

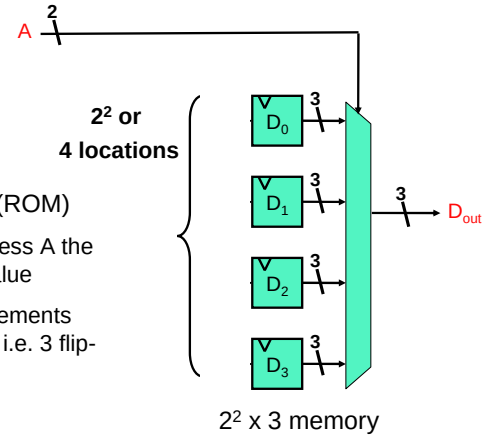
Addressability can vary

- We can also have manufacture memory such that each memory location hold 16-bit or 32-bit
- Example: LC3 has 2^{16} x 16(= 2^4) memory = 2^{17} x 2^3 = 2^{17} bytes

Sequential: Memory as a black box



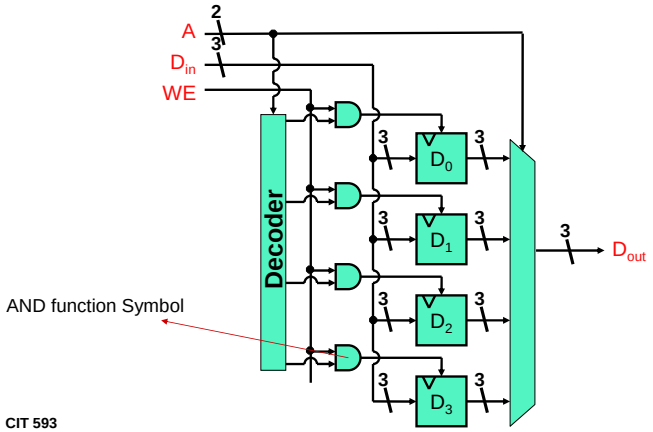
Memory (1 layer down)



Read Only Memory (ROM)

- Based on the address A the Mux choses the value
- D_{0-3} are storage elements each storing 3 bits i.e. 3 flip-flops

Memory – Read and Write



Sequential: Register File

Small fast memory

- **Small – only 8-32 locations. E.g. LC3 has only 8 registers**
- **Fast – For faster access compared to 256 MB memory**
  - ➢ **This all driven by technology and cost**

Used for temporary storage

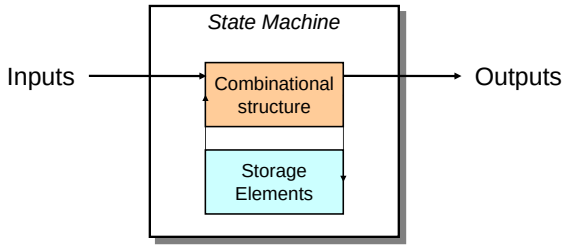
- **To avoid going back and forth to the memory**
- **For intermediate calculations**
  - ➢ **E.g. (A+B)*C**

Analogy: we use our desk to keep a few books (immediate need), and use a bookshelf to store the other books.

Sequential: State Machine

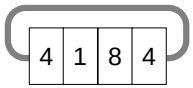
Another type of sequential structure

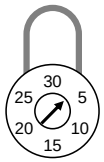
- **Combines combinational structure with storage (memory element)**
- **"Remembers" state (past information), and changes output (and state) based on inputs and current state**



Combinational vs. Sequential Example

Two types of "combination" locks



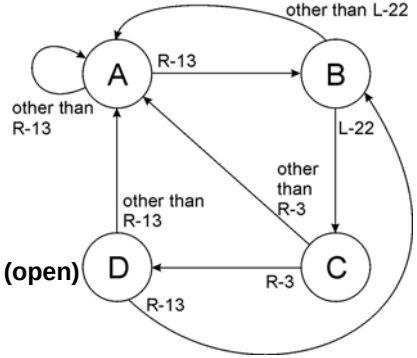


Combinational
Success depends only on the values, not the order in which they are set.

Sequential
Success depends on the sequence of values (e.g, R-13, L-22, R-3).

Sequential Lock State Diagram

Shows states and actions that cause a transition between states



Sequence
R-13, L-22, R-3

State of Sequential Lock

Our lock example has four different states, labeled A-D:

A: The lock is not open,
and no relevant operations have been performed

B: The lock is not open,
and the user has completed the R-13 operation

C: The lock is not open,
and the user has completed R-13, followed by L-22

D: The lock is open

Storage elements for sequential circuits

Recall that flip-flop/storage unit stores one bit

- **State can be 0 or 1**

Number of storage elements (flip-flops)

- **Determined by number of states (and representation of states)**

Examples

- **Sequential lock**
  - **Four states – two bits**

- **Basketball scoreboard**
  - **8 bits for each score, 6 bits for playtime minutes and seconds,1 bit for possession arrow,**

Next

Put all the components together and we have a working computer

- **Combinational Elements: Adder, Decoder**
  - **Forms the Arithmetic & Logic Unit a.k.a ALU which does all the processing**
- **Sequential Elements:**
  - **Memory: Registers (temp storage), Memory (storage while power is on)**
  - **Control a.k.a brain of the computer : Co-ordinate control signals and data movement to and from memory and I/O**