

Computer Architecture: Logic Design

CSci 2021: Machine Architecture and Organization
 October 26th-29th, 2018

Your instructor: Stephen McCamant

Based on slides originally by:
 Randy Bryant and Dave O'Hallaron

Overview of Logic Design

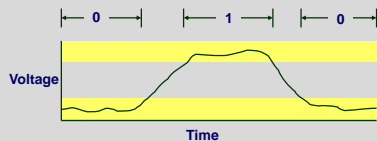
Fundamental Hardware Requirements

- Communication
 - How to get values from one place to another
- Computation
- Storage

Bits are Our Friends

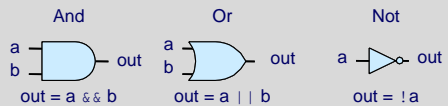
- Everything expressed in terms of values 0 and 1
- Communication
 - Low or high voltage on wire
- Computation
 - Compute Boolean functions
- Storage
 - Store bits of information

Digital Signals

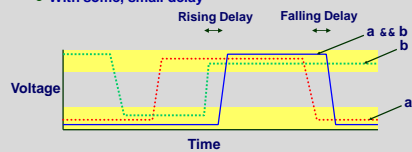


- Use voltage thresholds to extract discrete values from continuous signal
- Simplest version: 1-bit signal
 - Either high range (1) or low range (0)
 - With guard range between them
- Not strongly affected by noise or low quality circuit elements
 - Can make circuits simple, small, and fast

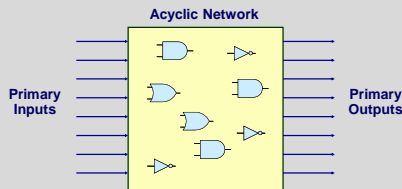
Computing with Logic Gates



- Outputs are Boolean functions of inputs
- Respond continuously to changes in inputs
 - With some, small delay



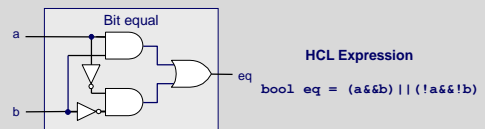
Combinational Circuits



Acyclic Network of Logic Gates

- Continuously responds to changes on primary inputs
- Primary outputs become (after some delay) Boolean functions of primary inputs

Bit Equality

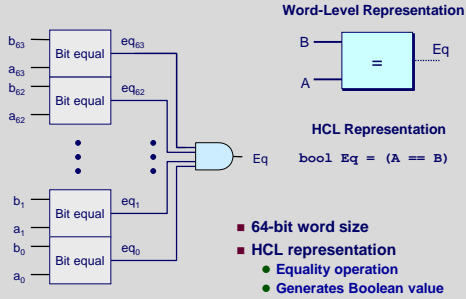


- Generate 1 if a and b are equal

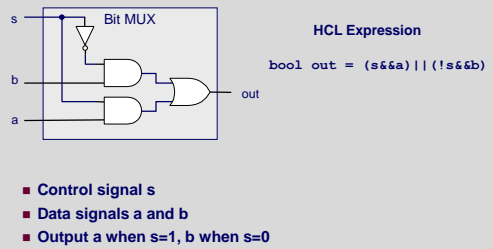
Hardware Control Language (HCL)

- Very simple hardware description language
 - Boolean operations have syntax similar to C logical operations
- We'll use it to describe control logic for processors

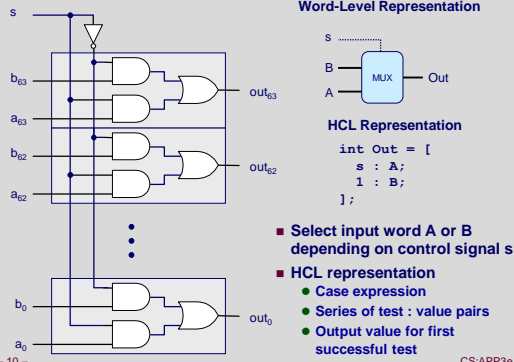
Word Equality



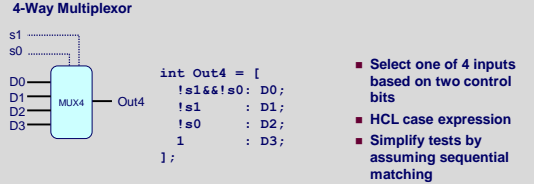
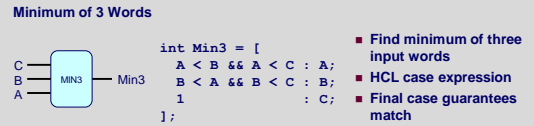
Bit-Level Multiplexor



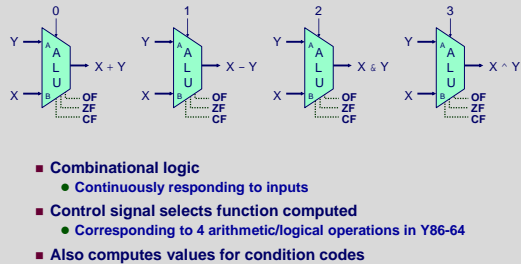
Word Multiplexor



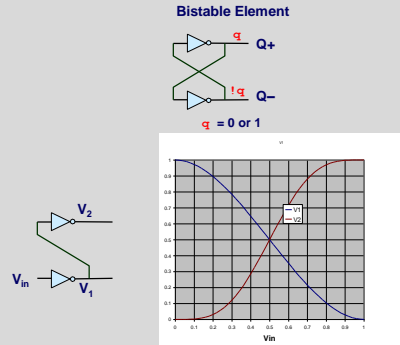
HCL Word-Level Examples



Arithmetic Logic Unit

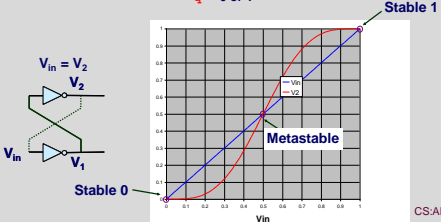
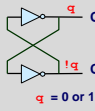


Storing 1 Bit

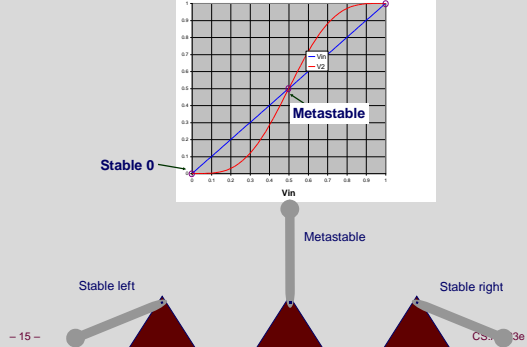


Storing 1 Bit (cont.)

Bistable Element

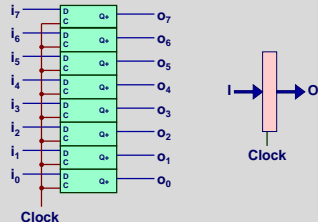


Physical Analogy



Registers

Structure



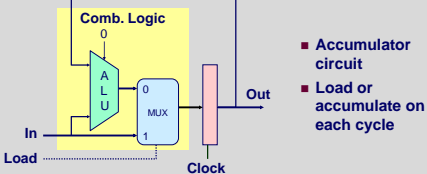
- Stores word of data
 - Different from *program registers* seen in assembly code
- Collection of edge-triggered latches
- Loads input on rising edge of clock

Register Operation

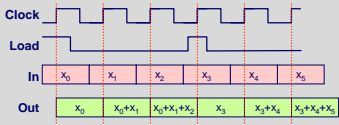


- Stores data bits
- For most of time acts as barrier between input and output
- As clock rises, loads input

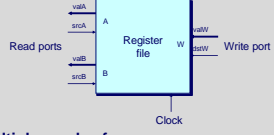
State Machine Example



- Accumulator circuit
- Load or accumulate on each cycle

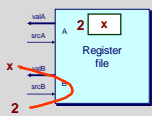


Random-Access Memory



- Stores multiple words of memory
 - Address input specifies which word to read or write
- Register file
 - Holds values of program registers
 - %rax, %rsp, etc.
 - Register identifier serves as address
 - » ID 15 (0xF) implies no read or write performed
- Multiple Ports
 - Can read and/or write multiple words in one cycle
 - » Each has separate address and data input/output

Register File Timing

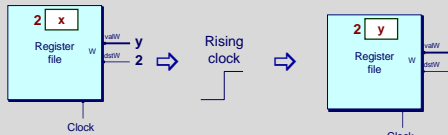


Reading

- Like combinational logic
- Output data generated based on input address
 - After some delay

Writing

- Like register
- Update only as clock rises



- 24 -

CS:APP3e

Hardware Control Language

- Very simple hardware description language
- Can only express limited aspects of hardware operation
 - Parts we want to explore and modify

Data Types

- bool: Boolean
 - a, b, c, ...
- int: words
 - A, B, C, ...
 - Does not specify word size—bytes, 64-bit words, ...

Statements

- `bool a = bool-expr ;`
- `int A = int-expr ;`

- 25 -

CS:APP3e

HCL Operations

- Classify by type of value returned

Boolean Expressions

- Logic Operations
 - `a && b, a || b, !a`
- Word Comparisons
 - `A == B, A != B, A < B, A <= B, A >= B, A > B`
- Set Membership
 - `A in { B, C, D }`
 - » Same as `A == B || A == C || A == D`

Word Expressions

- Case expressions
 - `[a : A; b : B; c : C]`
 - Evaluate test expressions a, b, c, ... in sequence
 - Return word expression A, B, C, ... for first successful test

- 26 -

CS:APP3e

Summary

Computation

- Performed by combinational logic
- Computes Boolean functions
- Continuously reacts to input changes

Storage

- Registers
 - Hold single words
 - Loaded as clock rises
- Random-access memories
 - Hold multiple words
 - Possible multiple read or write ports
 - Read word when address input changes
 - Write word as clock rises

- 27 -

CS:APP3e