

Midterm Review

CIT 595
Spring 2008

Exam Details

- Expect questions like on homeworks
- Bring a calculator
- Closed book and notes
- Information provided: instruction encoding, Flip-Flop characteristic equation etc..

CIT 595

2

Boolean Algebra and Comb. Logic

- Know symbols, and truth tables for NOT, AND, OR, NAND, NOR, XOR, XNOR
- Know the different Canonical forms: SOP, POS
- Boolean Reduction using KMap
- Know at least the Boolean Identities of the AND & OR and DeMorgans
- Apply all of the above to a problem

CIT 595

3

Example 1: NZP logic

- X is 3-bit 2's complement value, determine if it negative, zero, positive

X ₂	X ₁	X ₀	Z	N	P
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	0	1
0	1	1	0	0	1
1	0	0	0	1	0
1	0	1	0	1	0
1	1	0	0	1	0
1	1	1	0	1	0

CIT 595

4

Sequential Logic

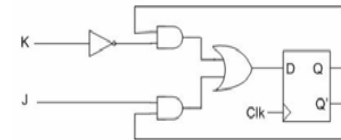
- Contains combinational logic + memory elements
- Again given some information, try to translate it into state diagram or state table

CIT 595

5

Example 2: D Flip-Flop to JK Flip-Flop

J	K	Q(t)	Q(t + 1) = D
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



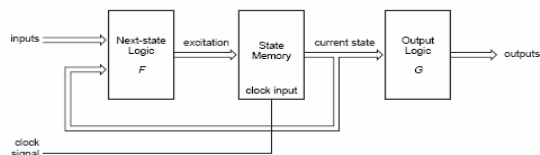
		K Q(t)			
		00	01	11	10
J	0	0	1	0	0
	1	1	1	0	1

$$D = J Q'(t) + K' Q(t)$$

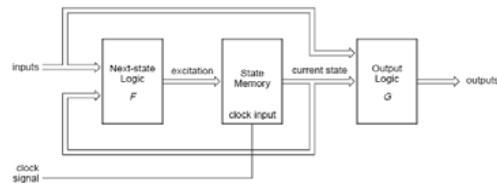
CIT 595

6

Finite State Machines



General Moore Machine



General Mealy Machine

CIT 595

7

FSM Example

Design a serial comparator for n-bit unsigned numbers x and y

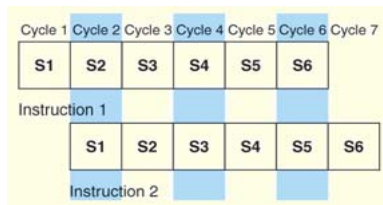
- The FSM takes two bits x_i and y_i at a time. Assume the two bits are being fed in from MSB to LSB.
 - E.g. x_{n-1} and y_{n-1} arrive at time unit 1 and x_{n-2} and y_{n-2} at time unit 2.
- The output of the FSM should be
 - 00 if the two values are equal
 - 10 if x has a larger value
 - 01 if y has a larger value
- Draw state-diagram

CIT 595

8

Pipelining

- Instruction cycle processing, each instruction takes some number of cycles
- Realize that while one phase of the instruction is active, the rest all are idle
- Pipelining exploits this idleness



CIT 595

9

Impact of Pipelining

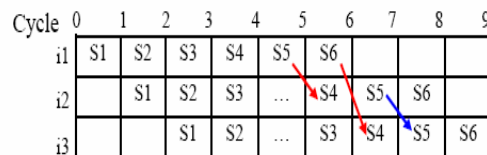
- Increase instruction throughput
- Ideal Conditions
 - CPI = 1
 - We find that ideally the speedup = # stages
 - Improve processor performance by increasing number of stages (i.e. shorten clock cycle time)
- Limitations due to Hazards
 - Structural
 - Data
 - Control

CIT 595

10

Pipelining Example (hw3 6b)

- i1: LDR R5, R1, #3
- i2: ADD R2, R5, #2
- i3: STR R2, R5, #3



CIT 595

11

CPI for Pipeline

Average CPI = \sum (CPI type x Fraction of Instructions per type)

CPI type = 1 + Additional Cycles

Additional Cycles for (others can get by due to forwarding):

- Branch
- Jumps
- Load

CIT 595

12

Hazard Resolution

- Stalls
- Forwarding
- Delayed Branching

Example:

```

LOOP ADD R6, R6, #1
      SUB R3, R3, R1
      STR R3, R6, #0
      BRp LOOP
    
```

CIT 595

13

CPU Performance

- Performance Metric: Response Time

$$\text{CPU Time} = \frac{\text{time}}{\text{program}} = \frac{\text{time}}{\text{cycle}} \times \frac{\text{cycles}}{\text{instruction}} \times \frac{\text{instructions}}{\text{program}}$$

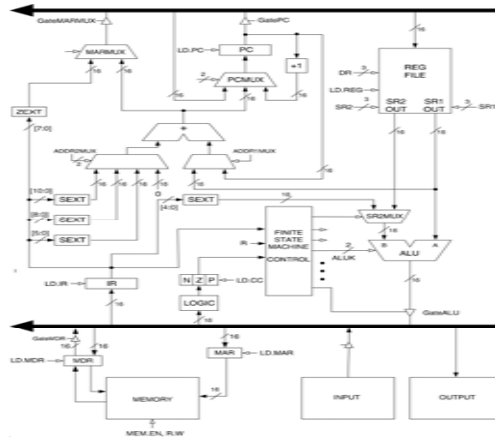
Clock Cycle time
CPI
Instruction Count

- Performance = 1/ Execution Time
- Performance Ratio/Speedup
 - A to B = Ex of B/ Ex of A

CIT 595

14

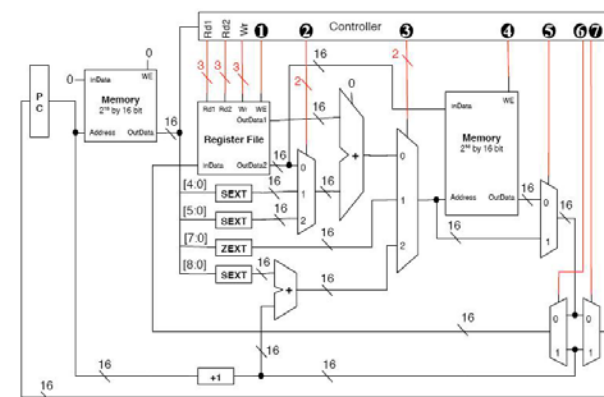
Microprogrammed LC3



CIT 595

15

Hardwired LC3



CIT 595

16

Control Signals in general

- There are some control signals that cannot be don't cares (XXX)
 - E.g. Signals that update register or memory
 - E.g. Gates that allow sharing of common bus
 - If one gate is on, the others must be off