Midterm Review

CIT 595
Spring 2007

Exam Details

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

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
- Apply all of the above to a problem

Example 1: NZP logic

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

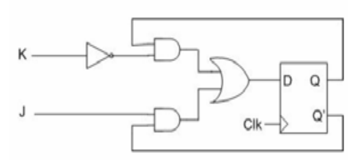
X_2	X_1	X_0	Z	N	P
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	0	1
0	1	1	0	0	1
1	0	0	0	1	0
1	0	1	0	1	0
1	1	0	0	1	0
1	1	1	0	1	0

Sequential Logic

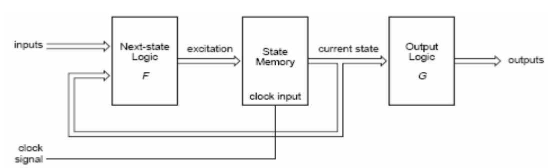
- Contains combinational logic + memory elements
- Again given some information, try to translate it into state diagram or state table

Example 2: D Flip-Flop to JK Flip-Flop (from quiz)

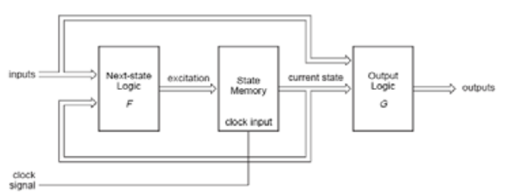
J	K	Q(t)	Q(t + 1) = D
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



Finite State Machines



General Moore Machine



General Mealy Machine

FSM Example

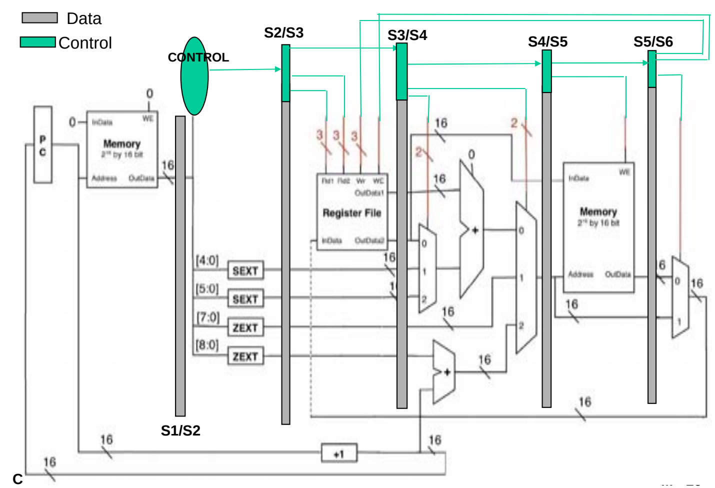
Designing a serial comparator for n-bit unsigned numbers x and y

- 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
  - output 00 if the two values are equal
  - output 10 if x has a larger value
  - output 01 if y has a larger value
- Draw state-diagram

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

LC3 pipeline



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

Pipelining Example 1 (hw3 6b)

```
i1:  ADD R5, R1, R3
i2:  LDR R2, R5, #2
i3:  STR R2, R6, #3
```

Cycle	0–1	1–2	2–3	3–4	4–5	5–6	6–7	7–8
i1	S1	S2	S3	S4	S5	S6		
i2		S1	S2	S3	S4	S5	S6	
i3			S1	S2	S3	S4	S5	S6

Pipelining Example 2 (hw3 6c)

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

Cycle 0 1 2 3 4 5 6 7 8 9

i1	S1	S2	S3	S4	S5	S6			
i2		S1	S2	S3	...	S4	S5	S6	
i3			S1	S2	...	S3	S4	S5	S6

CPI for Pipeline

Average CPI = Σ (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

Hazard Resolution

- Stalls
- Forwarding
- Delayed Branching

Example:

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


CPU Performance

- Performance Metric: Response Time
- **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
- Statistical Methods for performance Analysis
  - Arithmetic Mean
  - Weighted Average
  - Geometric Mean
  - Harmonic Mean

Geometric Mean (hw3 q4 in txt)

	Program	Execution Time on A	Execution Time Normalized to A	Execution Time on B	System B Normalized to A	Execution Time on C	System C Normalized to A
	v	45	1	125	0.36	75	0.60
	w	300	1	275	1.09	350	0.86
	x	250	1	100	2.50	200	1.25
	y	400	1	300	1.33	500	0.80
	z	800	1	1200	0.67	700	1.14
Geometric Mean			**1**		**0.97**		**0.89**
Arithmetic Mean		359		400		365	

The system is normalized to A.

For each program, **X/Normalized to A = Execution time of A / Execution time of X**. Therefore, if X/A is greater than 1 then X is better than A.

Geometric Ratio B/A = 0.97/1 = 0.97

Geometric Ratio C/A = 0.89/1 = 0.89

Geometric Ratio B/C = 0.97/0.89 = 1.089 (Same as when System B is Normalized to C)

Hence according to geometric mean ratio analysis, A is better than B. Similarly A is better than C and B is better than C.

According Arithmetic Mean, A is better than both B and C. However, C is better than B.

Memory

a. How many 256 × 8 RAM chips are needed to provide a memory capacity of 4096 bytes?
- 16

b. How many total bits are needed to access 4096 bytes
- 12

c. Size of the decoder per RAM chip?
- 8-to-256 decoder

d. Address 10 (A in hex) will be in which RAM chip according to:
- High-order interleaving - RAM chip # 0
- Low-order interleaving - RAM chip # 10