CS152 – Computer Architecture and Engineering

Lecture 4 – Overview of Logic Design

2003-09-04

Dave Patterson
(www.cs.berkeley.edu/~patterson)

www-inst.eecs.berkeley.edu/~cs152/

Review

- 4-LUT FPGAs are basically interconnect plus distributed RAM that can be programmed to act as any logical function of 4 inputs
- CAD tools do the partitioning, routing, and placement functions onto CLBs
- FPGAs offer compromise of performance, Non Recurring Engineering, unit cost, time to market vs. ASICs or microprocessors (plus software)

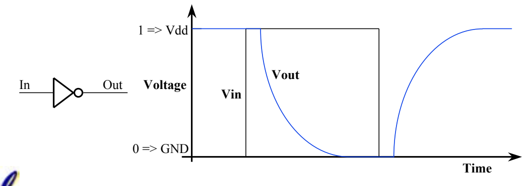
	Performance	*NRE*	*Unit Cost*	*TTM*
Better ↑	ASIC	MICRO	ASIC	MICRO
	FPGA	FPGA	MICRO	FPGA
Worse ↓	MICRO	ASIC	FPGA	ASIC

Outline

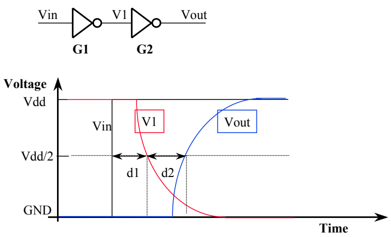
- Critical Path
- Setup / Hold Times
- Clock skew
- Finite State Machines
- One-hot encoding
- DeMorgan's Theorem
- Don't cares/ Karnaugh Maps
- IC costs
- FPGA CAD guided tour (if time permits)

Ideal versus Reality

- When input 0 -> 1, output 1 -> 0 but NOT instantly
  - Output goes 1 -> 0: output voltage goes from Vdd (5v) to 0v
- When input 1 -> 0, output 0 -> 1 but NOT instantly
  - Output goes 0 -> 1: output voltage goes from 0v to Vdd (5v)
- Voltage does not like to change instantaneously

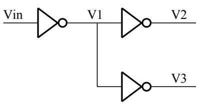


Series Connection



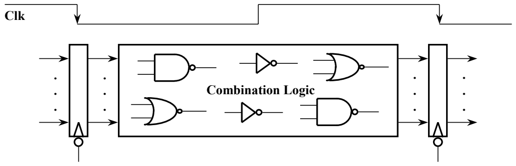
- Total Propagation Delay = Sum of individual delays = d1 + d2
- Capacitance C1 has two components:
  - Capacitance of the wire connecting the two gates
  - Input capacitance of the second inverter

Calculating Aggregate Delays; Critcal Path



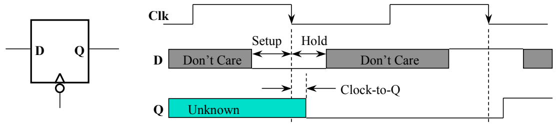
- Sum delays along serial paths
- Delay (Vin -> V2) ! = Delay (Vin -> V3)
  - Delay (Vin -> V2) = Delay (Vin -> V1) + Delay (V1 -> V2)
  - Delay (Vin -> V3) = Delay (Vin -> V1) + Delay (V1 -> V3)
- Capacitance$_1$ = Wire Capacitance
  + Capacitance$_{in}$ of Invertor$_2$
  + Capacitance$_{in}$ of Invertor$_3$
- Critical Path = The longest among the N parallel paths

Clocking Methodology



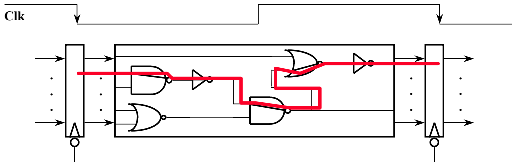
- All storage elements are clocked by the same clock edge
- The combination logic block's:
    - Inputs are updated at each clock tick
    - All outputs MUST be stable before the next clock tick

Storage Element's Timing Model



- Setup Time: Input must be stable BEFORE trigger clock edge
- Hold Time: Input must REMAIN stable after trigger clock edge
- Clock-to-Q time:
    - Output cannot change instantaneously at the trigger clock edge
    - Similar to delay in logic gates, two components:
        - Internal Clock-to-Q
        - Load dependent Clock-to-Q
- Typical for Virtex-E: ? Setup, ? Hold

Critical Path & Cycle Time

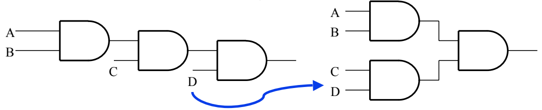


- Critical path: the slowest path between any two storage devices
- Cycle time is a function of the critical path
- must be greater than:

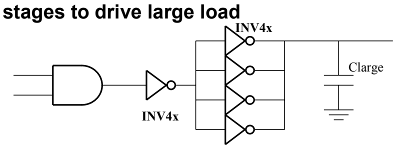
Clock-to-Q + Longest Path through Combination Logic + Setup

Tricks to Reduce Cycle Time

- Reduce the number of gate levels



- **Pay attention to loading**
    - **One gate driving many gates is a bad idea**
    - **Avoid using a small gate to drive a long wire**
- **Use multiple stages to drive large load**



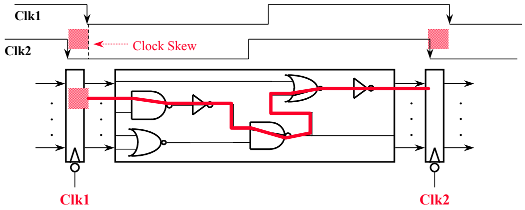
Administrivia (1/2)

- Even though there is lots of space in the class, there is a mistake in Telebears that is preventing us from adding people into sections. Therefore, if you are not enrolled in the course, but want to get in, here is what you have to do:
- By Friday 9/5 1:00pm, talk to Michael David Sasson (in person) in 379 Soda
- Be sure to tell him which section you want to move into. If you don't talk to him by Friday, we can't add you to the class!

Administrivia (2/2)

- Friday 9/5 lab demo in 125 Cory 3-4
    - If didn't take CS 150, should go to demo
- Office hours in Lab
    - Mon 5 – 6:30 Jack, Tue 3:30-5 Kurt, Wed 3 – 4:30 John
- Dave's office hours Tue 3:30 – 5
- HW #1 Due Wed 9/10
- Lab #2 done in pairs since 15 FPGA boards, 33 PCs. Due Monday 9/15
- Form 4 or 5 person teams by Friday 9/12

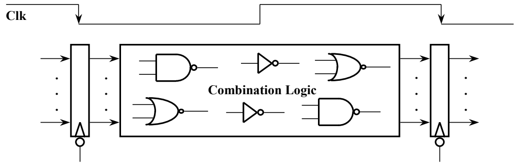
Clock Skew's Effect on Cycle Time



- The worst case scenario for cycle time consideration:
  - The input register sees CLK1
  - The output register sees CLK2
- Cycle Time - Clock Skew ≥ CLK-to-Q + Longest Delay + Setup
  ⇒ Cycle Time ≥ CLK-to-Q + Longest Delay + Setup + Clock Skew

---

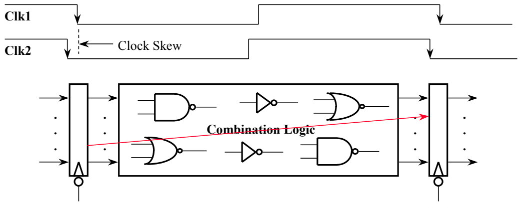
How to Avoid Hold Time Violation?



- Hold time requirement:
  - Input to register must NOT change immediately after the clock tick
- This is usually easy to meet in the "edge trigger" clocking scheme
- CLK-to-Q + Shortest Delay Path must be greater than Hold Time

---

Clock Skew's Effect on Hold Time



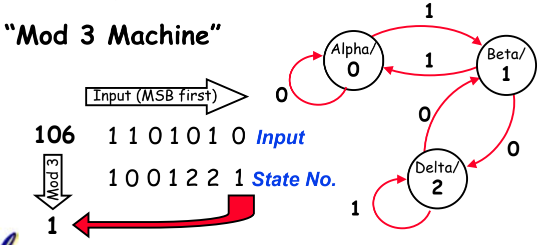
- The worst case scenario for hold time consideration:
  - The input register sees CLK2
  - The output register sees CLK1
  - fast FF2 output must not change input to FF1 for same clock edge
- (CLK-to-Q + Shortest Delay Path - Clock Skew) > Hold Time

Danger is one fast path (e.g., 1 gate) violate hold time

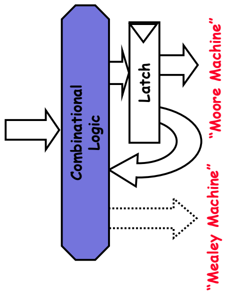
Finite State Machines:

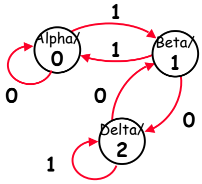
- System state is explicit in representation
- Transitions between states represented as arrows with inputs on arcs.
- Output may be either part of state or on arcs

"Mod 3 Machine"



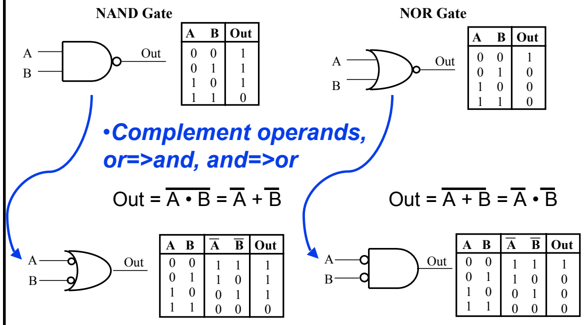
FSM Implementation as Combinational logic + Latch





Input	State_{old}	State_{new}
0	00	00
0	01	10
0	10	01
1	00	01
1	01	00
1	10	10

DeMorgan's theorem: Push Bubbles and Morph



A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0

A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

- ***Complement operands, or=>and, and=>or***

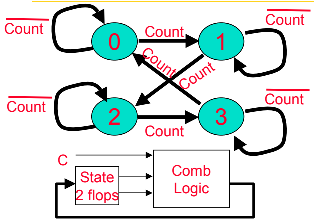
Out = $\overline{A \cdot B} = \overline{A} + \overline{B}$

Out = $\overline{A + B} = \overline{A} \cdot \overline{B}$

A	B	$\overline{A}$	$\overline{B}$	Out
0	0	1	1	1
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

A	B	$\overline{A}$	$\overline{B}$	Out
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

Example: Simplification of logic



S_1	S_0	C	S_1’	S_0’
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

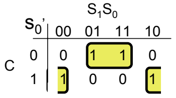
$$\begin{aligned} S_0' &= (\overline{S_1}\cdot\overline{S_0}\cdot C)+(\overline{S_1}\cdot S_0\cdot\overline{C})+(S_1\cdot\overline{S_0}\cdot C)+(S_1\cdot S_0\cdot\overline{C}) \\ &= (\overline{S_0}\cdot C)+(S_0\cdot\overline{C}) \end{aligned}$$

$$\begin{aligned} S_1' &= (\overline{S_1}\cdot S_0\cdot C)+(S_1\cdot\overline{S_0}\cdot\overline{C})+(S_1\cdot\overline{S_0}\cdot C)+(S_1\cdot S_0\cdot\overline{C}) \\ &= (\overline{S_1}\cdot S_0\cdot C)+(S_1\cdot\overline{C})+(S_1\cdot\overline{S_0}) \end{aligned}$$

Karnaugh Map for easier simplification

Group Ones in powers of 2, simplify, OR-together

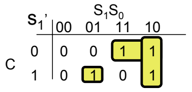
S_1	S_0	C	S_1’	S_0’
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0



S_1S_0

S_0’		00	01	11	10
C	0	0	1	1	0
	1	1	0	0	1

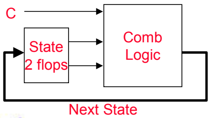
$$S_0' = (\overline{S_0}\cdot C)+(S_0\cdot\overline{C})$$



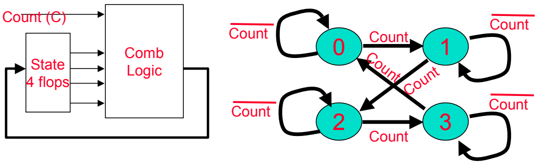
S_1S_0

S_1’		00	01	11	10
C	0	0	0	1	1
	1	0	1	0	1

$$S_1' = (\overline{S_1}\cdot S_0\cdot C)+(S_1\cdot\overline{C})+(S_1\cdot\overline{S_0})$$



“One-Hot Encoding”: 1 FF/state vs. $\log_2$ FFs



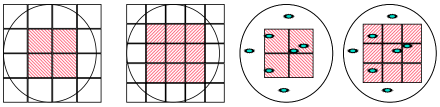
- One Flip-flop per state
  => no decode HW of state no.
- Only one state bit = 1 at a time
- Much faster combinational logic
- Tradeoff: Size ⇔Speed

$$S_0' = (S_0\cdot\overline{C})+(S_3\cdot C)$$
$$S_1' = (S_1\cdot\overline{C})+(S_0\cdot C)$$
$$S_2' = (S_2\cdot\overline{C})+(S_1\cdot C)$$
$$S_3' = (S_3\cdot\overline{C})+(S_2\cdot C)$$

Integrated Circuit Costs

$$\text{Die cost} = \frac{\text{Wafer cost}}{\text{Dies per Wafer} \ * \ \text{Die yield}}$$

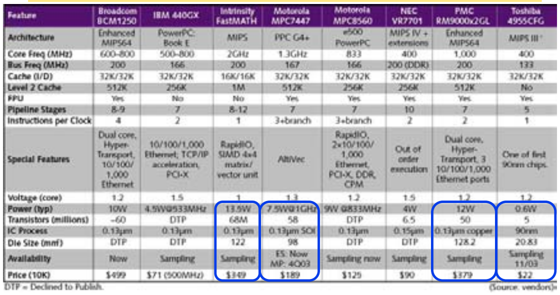
$$\text{Dies per wafer} = \frac{\pi * (\text{Wafer\_diam} / 2)^2}{\text{Die Area}} - \frac{\pi * \text{Wafer\_diam}}{\sqrt{2 * \text{Die Area}}} - \text{Test dies} \approx \frac{\text{Wafer Area}}{\text{Die Area}}$$



$$\text{Die Yield} = \frac{\text{Wafer yield}}{\left\{1 + \frac{\text{Defects\_per\_unit\_area} \ * \ \text{Die\_Area}}{\alpha}\right\}^{\alpha}}$$

Die Cost is roughly polynomial with die size: (die area)^3 or (die area)^4

Real World Examples: 2003



Feature	Broadcom BCM1250	IBM 440GX	Intrinsity FastMATH	Motorola MPC7447	Motorola MPC8560	NEC VR7701	PMC RM9000x2GL	Toshiba 4955CFG
Architecture	Enhanced MIPS64	PowerPC: Book E	MIPS	PPC G4+	e500 PowerPC	MIPS IV + extensions	Enhanced MIPS64	MIPS III'
Core Freq (MHz)	600-800	500-800	2GHz	1.3GHz	833	400	1,000	400
Bus Freq (MHz)	200	166	200	167	166	200 (DDR)	200	133
Cache (I/D)	32K/32K	32K/32K	16K/16K	32K/32K	32K/32K	32K/32K	32K/32K	32K/32K
Level 2 Cache	512K	256K	1M	512K	256K	256K	512K	No
FPU	Yes	No	No	Yes	Yes	Yes	Yes	Yes
Pipeline Stages	8-9	7	8-12	7	7	10	7	5
Instructions per Clock	4	2	1	3+branch	3+branch	2	2	1
Special Features	Dual core, Hyper-Transport, 10/100/1,000 Ethernet	10/100/1,000 Ethernet; TCP/IP acceleration, PCI-X	RapidIO, SIMD 4x4 matrix/vector unit	AltiVec	RapidIO, 2x10/100/1,000 Ethernet, PCI-X, DDR, CPM	Out of order execution	Dual core, Hyper-Transport, 3 10/100/1,000 Ethernet ports	One of first 90nm chips
Voltage (core)	1.2	1.5	1	1.3	1.2	1.5	1.2	1.2
Power (typ)	10W	4.5W@533MHz	13.5W	7.5W@1GHz	9W @833MHz	4W	12W	0.6W
Transistors (millions)	~60	DTP	68M	58	DTP	6.5	50	5
IC Process	0.13µm	0.13µm	0.13µm	0.13µm SOI	0.13µm	0.15µm	0.13µm copper	90nm
Die Size (mm²)	DTP	DTP	122	98	DTP	DTP	128.2	20.83
Availability	Now	Sampling	Sampling	ES: Now MP: 4Q03	Sampling now	Sampling	Sampling	Sampling 11/03
Price (10K)	$499	$71 (500MHz)	$349	$189	$125	$90	$379	$22

DTP = Declined to Publish.

(Source: vendors)

From "Chart Watch: Embedded Processors," *Microprocessor Report*, August 25, 2003

Die Yield : 1995 data

Raw Dice Per Wafer

wafer diameter	***die area (mm^2)***					
	100	***144***	***196***	***256***	***324***	***400***
6”/15cm	139	90	62	44	32	23
8”/20cm	265	177	124	90	68	52
10”/25cm	431	290	206	153	116	90
die yield	23%	19%	16%	12%	11%	10%

typical CMOS process: α =2, wafer yield=90%, defect density=2/cm2, 4 test sites/wafer

Good Dice Per Wafer (Before Testing!)

6”/15cm	31	16	9	5	3	2
8”/20cm	59	32	19	11	7	5
10”/25cm	96	53	32	20	13	9

typical cost of an 8”, 4 metal layers, 0.5um CMOS wafer: ~$2000

Real World Examples: 1993

Chip	*Metal layers*	*Line width*	*Wafer cost*	*Defect /cm²*	*Area mm²*	*Dies/ wafer*	*Yield*	*Die Cost*
386DX	2	0.90	$900	1.0	43	360	71%	$4
486DX2	3	0.80	$1200	1.0	81	181	54%	$12
PowerPC 601	4	0.80	$1700	1.3	121	115	28%	$53
HP PA 7100	3	0.80	$1300	1.0	196	66	27%	$73
DEC Alpha	3	0.70	$1500	1.2	234	53	19%	$149
SuperSPARC	3	0.70	$1700	1.6	256	48	13%	$272
Pentium	3	0.80	$1500	1.5	296	40	9%	$417

From "Estimating IC Manufacturing Costs," by Linley Gwennap, *Microprocessor Report*, August 2, 1993, p. 15

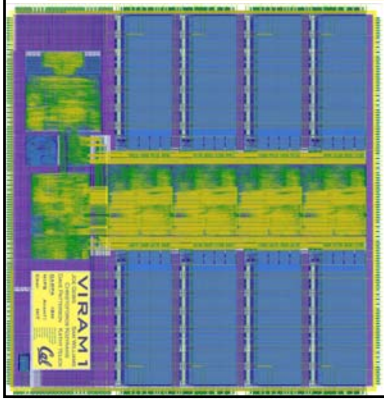
Other Costs

$$\text{IC cost} = \frac{\text{Die cost} + \text{Testing cost} + \text{Packaging cost}}{\text{Final test yield}}$$

Packaging Cost: depends on pins, heat dissipation

Chip	**Die cost**	**Package**			**Test & Assembly**	**Total**
		pins	**type**	**cost**		
386DX	**$4**	**132**	**QFP**	**$1**	**$4**	**$9**
486DX2	**$12**	**168**	**PGA**	**$11**	**$12**	**$35**
PowerPC 601	**$53**	**304**	**QFP**	**$3**	**$21**	**$77**
HP PA 7100	**$73**	**504**	**PGA**	**$35**	**$16**	**$124**
DEC Alpha	**$149**	**431**	**PGA**	**$30**	**$23**	**$202**
SuperSPARC	**$272**	**293**	**PGA**	**$20**	**$34**	**$326**
Pentium	**$417**	**273**	**PGA**	**$19**	**$37**	**$473**

Berkeley Vector IRAM chip: 2003



- MIPS CPU + multimedia coprocessor + 104 Mbits (13 Mbytes) of on-chip main memory
- 2 W, 200 MHz
- Die size: 325 mm², in 0.18 micron, 125M transistors
- App: future PDAs

Peer Instruction

- Which of the following are false?

1. $\overline{(A + B) \bullet (C + D)} = (\overline{A} \bullet \overline{B}) + (\overline{C} + \overline{D})$
2. Clock skew can lead to setup time violations
3. Clock skew can lead to hold time violations
4. If the die size is doubled, the cost of the die is roughly quadrupled

---

In conclusion, Logic review …

- Critical Path is longest among N parallel paths
- Setup Time and Hold Time determine how long Input must be stable before and after trigger clock edge
- Clock skew is difference between clock edge in different parts of hardware; it affects clock cycle time and can cause hold time, setup time violations
- FSM specify control symbolically
  - Moore machine easiest to understand, debug
  - "One shot" reduces decoding for faster FSM
- Die size affects both dies/wafer and yield