



Exam Review

Writing Exams

Asynchronous Circuits

Races, Cycles and State Assignment

Synchronous Circuits

State-Graph Construction

and Small Problems

Also

Multiple Outputs,

and a Harder Combinational Problem

Examples



CARLETON UNIVERSITY

April 2006

Final

Name: _____

Number: _____

Signature: _____

DURATION: 3 HOURS No. of Students: 219

Department Name & Course Number: Electronics ELEC 2607 A,B and C

Course Instructor(s) T.G.Ray and J. Knight

AUTHORIZED MEMORANDA **TURN OFF** cell phones and personal communications equipment and **LEAVE THEM AT THE FRONT.** Notes, books, and non-communicating calculator are allowed.

Students **MUST** count the number of pages in this examination question paper **before beginning to write**, and report any discrepancy immediately to a proctor. This question paper has **10** pages.

This examination paper **May Not** be taken from the examination room.

In addition to this question paper, students require: an examination booklet yes no *may request*
a Scantron sheet yes no

Please answer on the examination paper. The correct answers should easily fit in the space given.
You may ask for a booklet if you need it.

1 Boolean (1% for stating rules and map names.)

1%

If you use a map, indicate in the space underneath it, for which function it is.

If you use algebra, indicate the rules used at the right side of each line.

Note: 1% wasn't enough last year.

map of _____



Writing Exams

Timing yourself: 3 hours, 100% 1.5 min for 1% or 15 min for 10%.
1/2 hour for cleanup.

Give Up: If you can't get it move on.

RTFQ: Read The Foolish Question; read it again DAMN IT
What was asked for?
Just the state table? Then don't make K-maps.
Did it want a circuit. Then draw it!

Look for simple methods:

Don't do all algebra questions by multiplying out.

$$(X+A+B)(X+A+C)(BC+A) =$$

First look for simplifications.

Is there one inch space for the answer?

Do you need fifteen? Are you sure.

Simplify at each step:

Right elbow: $X + \text{anything} \cdot X = X$

Left elbow: $X + \text{anything} \cdot \bar{X} = X + \text{anything}$

Nose: $X + \bar{X} = 1$

$$F = (X+A+B)(X+A+C)(BC+A)$$

$$F_{\text{dual}} = XAB + XAC + (B+C)A$$

$$= XAB + XAC + BA + CA$$

$$= BA + CA$$

$$(F_{\text{dual}})_{\text{dual}} = F = (B+A)(C+A)$$

Take dual

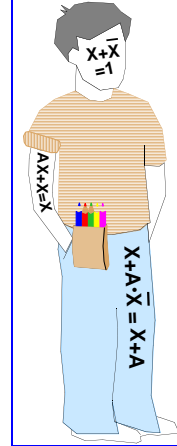
Dist 1

BA + anything \cdot BA = BA (use simp twice)

© John Knight

Electronics Department, Carleton University

Exam Ready



April 10, 2007



Writing Exams

Timing yourself: 3 hours, 100% 1.5 min for 1% or 15 min for 10%.
1/2 hour for cleanup.

Give Up: If you can't get it move on.

RTFQ: Read The Foolish Question; read it again DAMN IT
What was asked for?
Just the state table? Then don't make K-maps.
Did it want a circuit. Then draw it!

Look for simple methods:

Don't do all algebra questions by multiplying out.

$$(X+A+B)(X+A+C)(BC+A) = XXABC + XXA + XABC + XAA + XCBC + XCA + AXBC + AXA + AABC +$$

First look for simplifications.

Is there one inch space for the answer?

Do you need fifteen? Are you sure.

Simplify at each step:

Right elbow: $X + \text{anything} \cdot X = X$

Left elbow: $X + \text{anything} \cdot \bar{X} = X + \text{anything}$

Nose: $X + \bar{X} = 1$

$$F = (X+A+B)(X+A+C)(BC+A)$$

$$F_{\text{dual}} = XAB + XAC + (B+C)A$$

$$= XAB + XAC + BA + CA$$

$$= BA + CA$$

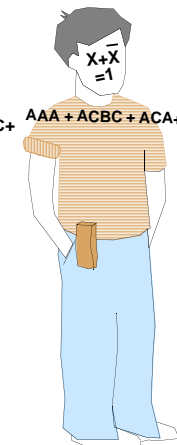
Take dual

Dist 1

BA + anything \cdot BA = BA (use simp twice)

© John Knight

Electronics Department, Carleton University



April 10, 2007



Find Races In Asynchronous State Tables

Example.1-1 .Find Races

a) State Tables for despicable machines

Present State FG	Next State F ⁺ G ⁺ For inputs:			
	ab = 00	ab = 01	ab = 11	ab = 10
FG=00	11	01	00	00
FG=01	11	10	01	01
FG=11	11	11	11	01
FG=10	10	11	00	10

b) Circle stable states

Present State FG	Next State F ⁺ G ⁺ For inputs:			
	ab = 00	ab = 01	ab = 11	ab = 10
FG=00	11	01	00	00
FG=01	11	10	01	01
FG=11	11	11	11	01
FG=10	10	11	00	10

c) Starting at stable states
Change one input, either a or b
Check for both F⁺ and G⁺ changing.

Present State FG	Next State F ⁺ G ⁺ For inputs:			
	ab = 00	ab = 01	ab = 11	ab = 10
FG=00	11	01	00	00
FG=01	11	10	01	01
FG=11	11	11	11	01
FG=10	10	11	00	10

d) Check race 1

Present State FG	Next State F ⁺ G ⁺ For inputs:			
	ab = 00	01	11	ab = 10
FG=00	11	01	00	00
FG=01	11	10	01	01
FG=11	11	11	11	01
FG=10	10	11	00	10

Race 1

Starts at
FG=00, ab=10
a changes 1->0
May end in
FG=11, ab=00
FG=10, ab=00

Double input changes never happen.

Races that don't start in a stable state are "who cares" unless part of a larger path.



Find Races In Asynchronous State Tables (Cont)

Example.1-2 .Find Races Continued

c) Starting at stable states
Change one input, either a or b
Check for both F⁺ and G⁺ changing.

Present State FG	Next State F ⁺ G ⁺ For inputs:			
	ab = 00	ab = 01	ab = 11	ab = 10
FG=00	11	01	00	00
FG=01	11	10	01	01
FG=11	11	11	11	01
FG=10	10	11	00	10

e) Check Race 2

Present State FG	Next State F ⁺ G ⁺ For inputs:			
	00	ab = 01	ab = 11	10
FG=00	11	01	00	00
FG=01	11	10	01	01
FG=11	11	11	11	01
FG=10	10	11	00	10

Fixed thanks to Priti

Race 2

Starts at
FG=01, ab=11
a changes 1->0
Will end in
FG=11, ab=01
Noncritical race

f) If the first step is not a race.
Check for both F⁺ and G⁺ changing
as transitions go down the column.

Present State FG	Next State F ⁺ G ⁺ For inputs:			
	ab = 00	ab = 01	ab = 11	ab = 10
FG=00	11	01	00	00
FG=01	11	10	01	01
FG=11	11	11	11	01
FG=10	10	11	00	10

g) Check Race 3

Present State FG	Next State F ⁺ G ⁺ For inputs:			
	00	ab = 01	ab = 11	10
FG=00	11	01	00	00
FG=01	11	10	01	01
FG=11	11	11	11	01
FG=10	10	11	00	10

Race 3

Starts at
FG=00, ab=11
a changes 1->0
Race involves
FG=01, ab=01
FG=10, ab=01
FG=11, ab=01
Will end in
FG=11, ab=01

Noncritical race between transient states



Finding Nastys In Asynchronous State Tables

Example.1-3 .Find Races

a) Find races

XY	x ⁺ y ⁺ for inputs ab:			
	00	01	11	10
00	00	01	10	00
01	01	11	00	10
11	00	00	01	11
10	00	10	11	10

b) Circle stable states

XY	x ⁺ y ⁺ for inputs ab:			
	00	01	11	10
00	00	01	10	00
01	01	11	00	10
11	00	00	01	11
10	00	10	11	10

c) Starting from stable states look for double state changes

XY	x ⁺ y ⁺ for inputs ab:			
	00	01	11	10
00	00	01	10	00
01	01	11	00	10
11	00	00	01	11
10	00	10	11	10

Not allowed input change
Doesn't start in a stable state

d) Check first race

XY	x ⁺ y ⁺ for inputs ab:			
	00	01	11	10
00	00	01	10	00
01	01	11	00	10
11	00	00	01	11
10	00	10	11	10

Critical.

Can end in three different different states

Start state 01 (ab=00)
End: one of state 00 (ab=10) state 11 (ab=10) state 10 (ab=00)

e) Check second race

XY	x ⁺ y ⁺ for inputs ab:			
	00	01	11	10
00	00	01	10	00
01	01	11	00	10
11	00	00	01	11
10	00	10	11	10

Critical.

Can end in two different states

Start state 11 (ab=10)
End: one of state 00 (ab=00) state 01 (ab=00)

Remember double input changes are not allowed.

Races that don't start in a stable state are "who cares" unless part of a larger path.



Finding More Nastys In Asynchronous State Tables

Example.1-4 Find Noncritical Races

One entry is changed from previous state table.

a) Find a noncritical races

XY	x ⁺ y ⁺ for inputs ab:			
	00	01	11	10
00	00	01	10	00
01	01	10	00	10
11	00	00	01	11
10	00	10	11	10

changed

b) Circle stable states

XY	x ⁺ y ⁺ for inputs ab:			
	00	01	11	10
00	00	01	10	00
01	01	10	00	10
11	00	00	01	11
10	00	10	11	10

c) Starting from stable states look for races

XY	x ⁺ y ⁺ for inputs ab:			
	00	01	11	10
00	00	01	10	00
01	01	10	00	10
11	00	00	01	11
10	00	10	11	10

d) Check new race

XY	x ⁺ y ⁺ for inputs ab:			
	00	01	11	10
00	00	01	10	00
01	01	10	00	10
11	00	00	01	11
10	00	10	11	10

Race.

May go to three states.

Start in 01 ab=00->01

e) Check secondary paths

XY	x ⁺ y ⁺ for inputs ab:			
	00	01	11	10
00	00	01	10	00
01	01	10	00	10
11	00	00	01	11
10	00	10	11	10

But from any state we soon head back to 10.

We will eventually reach state 10 (ab=01).

Noncritical Race.



Finding Still More Nastys

Example.1-5 Find a True Cycle

One which oscillates until an input changes..

a) Find a true cycle

XY	x ⁺ y ⁺ for inputs ab:			
	00	01	11	10
00	00	01	10	00
01	01	10	00	10
11	00	00	01	11
10	00	10	11	10

b) Circle stable states

XY	x ⁺ y ⁺ for inputs ab:			
	00	01	11	10
00	00	01	10	00
01	01	10	00	10
11	00	00	01	11
10	00	10	11	10

c) Look down columns for cycles

XY	x ⁺ y ⁺ for inputs ab:			
	00	01	11	10
00	00	01	10	00
01	01	10	00	10
11	00	00	01	11
10	00	10	11	10

Cycle
No stable state
Oscillates until
input changes.

d) Check for other cycles

XY	x ⁺ y ⁺ for inputs ab:			
	00	01	11	10
00	00	01	10	00
01	01	10	00	10
11	00	00	01	11
10	00	10	11	10

Not a
cycle
eventually
goes to
10



Finding Still More Nastys

Example.1-6 Find a False Cycle **NOT DONE IN 2007**

One which dies out quickly..

a) Circle stable states

b) Find cycle that dies out

XY	x ⁺ y ⁺ for inputs ab:			
	00	01	11	10
00	00	01	10	00
01	01	11	00	10
11	00	00	01	11
10	00	10	11	10

c) Look down columns for cycles

XY	x ⁺ y ⁺ for inputs ab:			
	00	01	11	10
00	00	01	10	00
01	01	11	00	10
11	00	00	01	11
10	00	10	11	10

Changed from last slide

d) Cycle has a race

XY	x ⁺ y ⁺ for inputs ab:			
	00	01	11	10
00	00	01	10	00
01	01	11	00	10
11	00	00	01	11
10	00	10	11	10

The race can
take us to 10
which is stable.

Eventually it
will go to 10
It won't take long
usually.

If a cycle has a race which can send it to a stable state, it will go to that stable state.



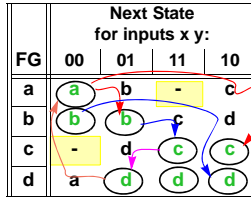
Mildly Difficult State Assignments

Example.1-7 Find a Race Free State Assignment
Simultaneous input changes never happen.

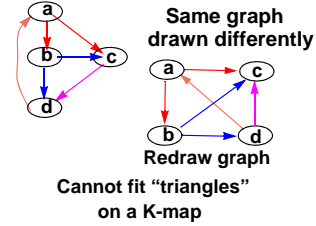
a) Asynchronous State Table
Circle Stable States

FG	Next State for inputs x y:			
	00	01	11	10
a	a	b	-	c
b	b	b	c	d
c	-	d	c	c
d	a	d	d	d

b) Show allowed transitions



c) Make graph of transitions.



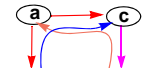
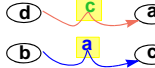
"don't cares" are "-"

d) Look at using "-"

FG	Next State for inputs x y:			
	00	01	11	10
a	a	b	c	c
b	b	b	a	d
c	a	d	c	c
d	c	d	d	d

Fill in the don't cares

e) Graph transitions through "-"
"-" become transient states



New graph with "-" used

f) Assign States

	0	1
0	a	c
1	b	d

Possible assignment

g) Completed table

FG	Next state for inputs x y:			
	00	01	11	10
a=00	00	01	10	10
b=01	01	01	00	11
d=11	10	11	11	11
c=10	00	11	10	10



Another State Assignments

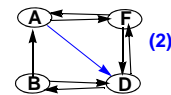
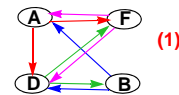
Example.1-8 Find a Race Free State Assignment
Simultaneous input changes never happen.

a): Asynchronous State Table;
Circle Stable States

Present State Q_1Q_2	Next State $Q_1^+Q_2^+$				Output Z
	xy=00	xy=01	xy=11	xy=10	
A=00	F	A	D	A	0
B	B	B	D	A	0
D	B	F	D	D	1
F	F	F	D	A	1

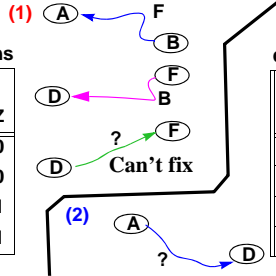
b): Draw Transitions

Present State Q_1Q_2	Next State $Q_1^+Q_2^+$				Output Z
	xy=00	xy=01	xy=11	xy=10	
A=00	F	A	D	A	0
B	B	B	D	A	0
D	B	F	D	D	1
F	F	F	D	A	1



With assignment (1)

Transitions that must be 2 step



With assignment (2)

c): Look at two-step transitions

Q_1Q_2	Next State for inputs x y:				Z
	00	01	11	10	
A	F	A	D	A	0
B	B	B	D	A	0
D	B	F	D	D	1
F	F	F	D	A	1

Will not work

d): Look at two-step transitions

Q_1Q_2	Next State for inputs x y:				Z
	00	01	11	10	
A	F	A	D	A	0
B	B	B	D	A	0
D	B	F	D	D	1
F	F	F	D	A	1

Works

e) Assign States

	0	1
0	A	F
1	B	D



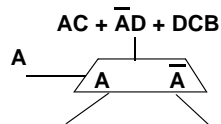
Common Mistakes

1. Saying ~~$\overline{a}b$ is the same as $\overline{a \cdot b}$~~
2. Saying an expression is equal to its dual.
3. Not using $AB + A = A$ to simplify expressions before using complex rules.
Not reducing using $A + \overline{A}E = A + E$.
Simplifying and reducing first saves algebra.
4. Saying ~~$X + 1 = X$~~
Everybody knows better than this, but they still do it.
5. When you take dual, or general Demorgan, do not put in the brackets in your head.
 ~~$(AB + C) + DE \implies (A+BC)(D+E)$~~
 $(AB + C) + DE \implies ((AB) + C) + (DE) \implies ((A+B)C)(D + E)$
6. Not knowing D2. $X + AB = (X + A)(X + B)$
7. A Karnaugh map may not give the simplest circuit,
but it does give the simplest Σ of Π circuit. *Unless you mess up the circles*



Boolean Algebra

1. Change to Σ of Π $(1 + Z)A\overline{X}$
2. Simplify $\overline{CD} + C$
3. Take the dual of $F = (A + B)(X + Y) + \overline{Z}$
4. Factor $X + B\overline{C}$
5. Factor $X + AB\overline{C}$
6. Find the dual of $G = (A + 1)(B + C) + \overline{D}$
7. Factor $AB + CD$
8. Construct simplest circuit with MUXs





Boolean Algebra

1. Change to Σ of Π	$(1 + Z)A\bar{X}$	$A\bar{X}$	rule $1+Z=Z$
2. Simplify	$\bar{C}D + C$	$\bar{C} + \bar{D} + C = 1$	rule DeMorg rule $C + \bar{C} = 1$
3. Take the dual of	$F = (A + B)(X + Y) + \bar{Z}$	$F = \{(A + B)(X + Y)\} + \bar{Z}$	bracket ANDs $F_{dual} = \{(AB) + (XY)\}\bar{Z}$
4. Factor	$X + B\bar{C}$	$(X + B)(X + \bar{C})$	rule D2
5. Factor	$X + AB\bar{C}$	$(X + AB)(X + \bar{C})$	rule D2 rule D2
6. Find the dual of	$G = (A + 1)(B + C) + \bar{D}$	$\{(A + 1)(B + C)\} + \bar{D}$	bracket ANDs $G_{dual} = \{(A0) + (BC)\}\bar{D}$
		$= B\bar{C}\bar{D}$	dual $1 \leftrightarrow 0$ rules $A0=0; x+0=x$
7. Factor	$AB + CD$	$(AB + C)(AB + D)$	rule D2
		$= (A + C)(B + C)(A + D)(B + D)$	rule D2
8. Construct with MUXs	$AC + \bar{A}D + DCB$ 		



Multiple Outputs

Example.1-9 Find the equations with minimum logic

Minimization with sharing

(1) Half Maps
Look for half-map circles (one letter terms)
These do not require an AND gate.
and can be circled without loss of potential gate sharing.

(2) No Friends (Lone "1") Rule
Circle squares that are "1" on only one map thus cannot be shared.

(2a) No Friends, They Went To The Dark Side, Rule
Find squares that cannot be usefully shared because that square on other maps is already circled. Circle them



Examp.1-9 Find the minimum Σ of Π

Minimum number of AND terms

Poor Method

Largest circles
7 AND terms

Better Method

(1) Circle half maps (none)

(2) No Friends, ("1"s on only one map)

Circle lone "1" now, they will never share

Unfortunately you have choices;
several ways to circle some squares

Choices



Find the minimum Σ of Π (Cont)

Choices

Choice (1)
Chosen circles are shaded

Shows unnecessary redundancy

John's Solution
6 AND terms

Choice (2)

Tom's Solution
5 AND terms

Choice (3)

Candice's Solution
5 AND terms



Find the minimum logic (Cont)

Another choice

John's Solution
6 AND terms

Your Solution
5 AND terms



Design of State Graphs

Example.1-10 A **1101** or **1010** Machine

Detect **1101** or **1010** with overlap. Output should last for a full clock cycle.
First treat as individual machines.

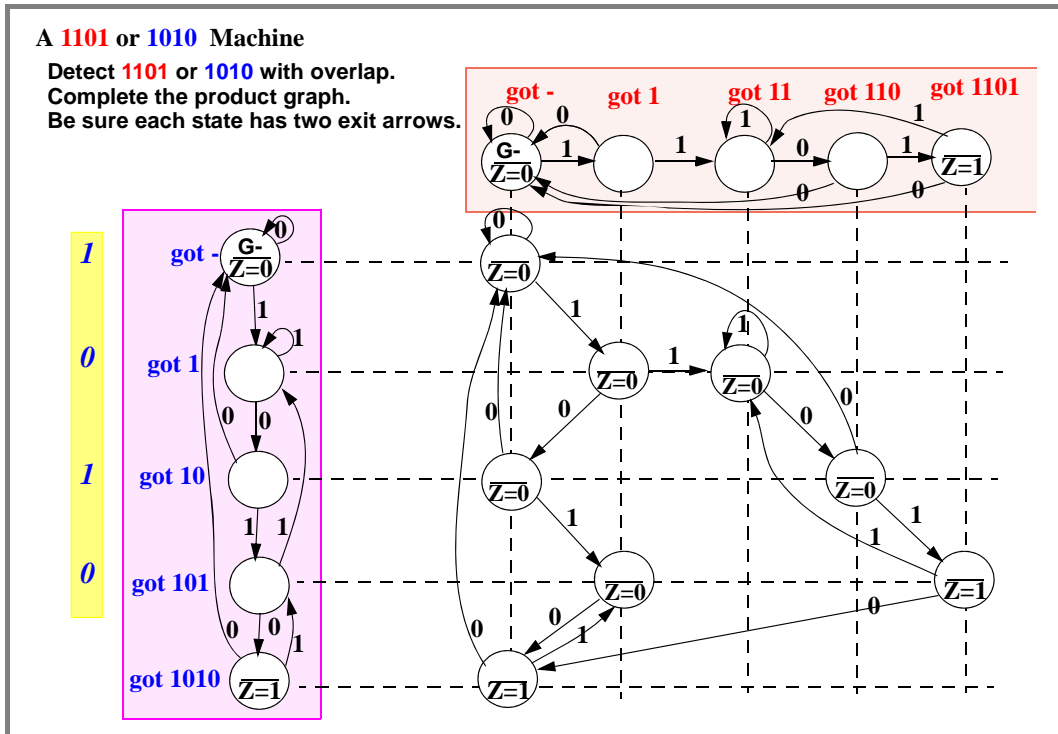
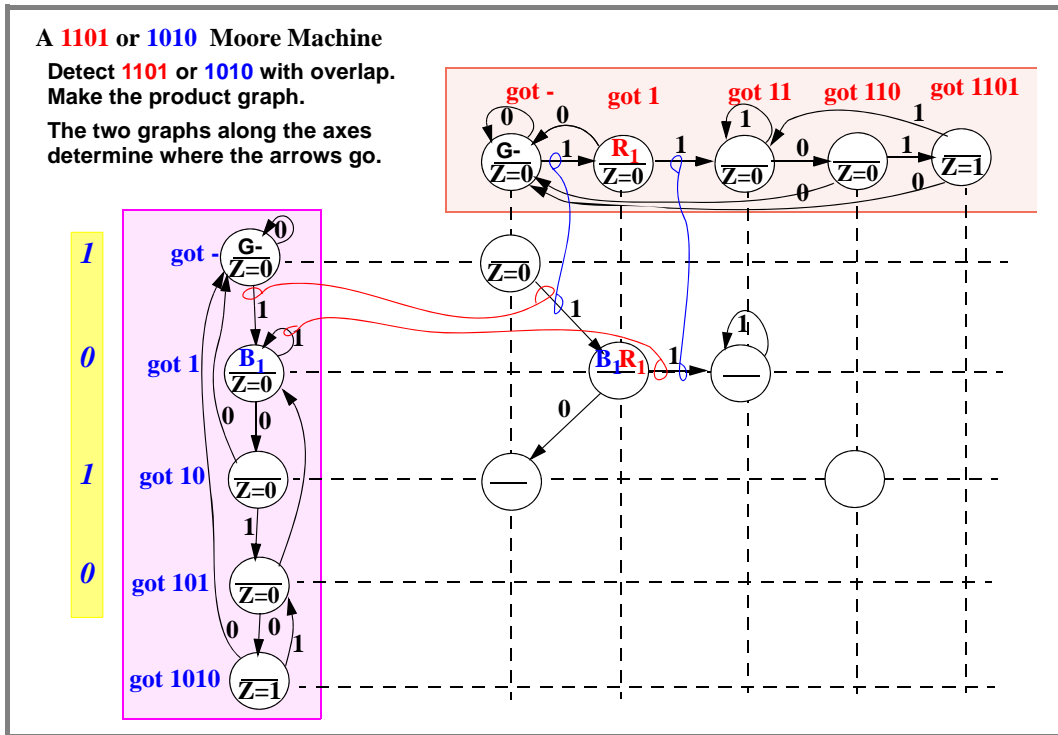
Means Moore

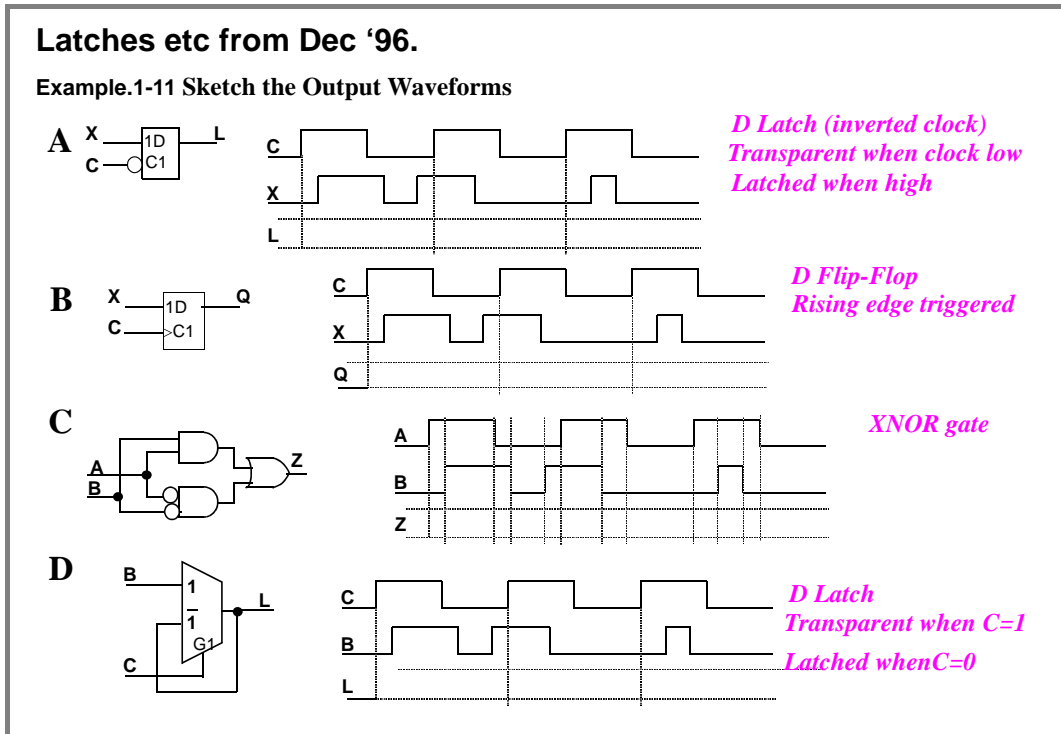
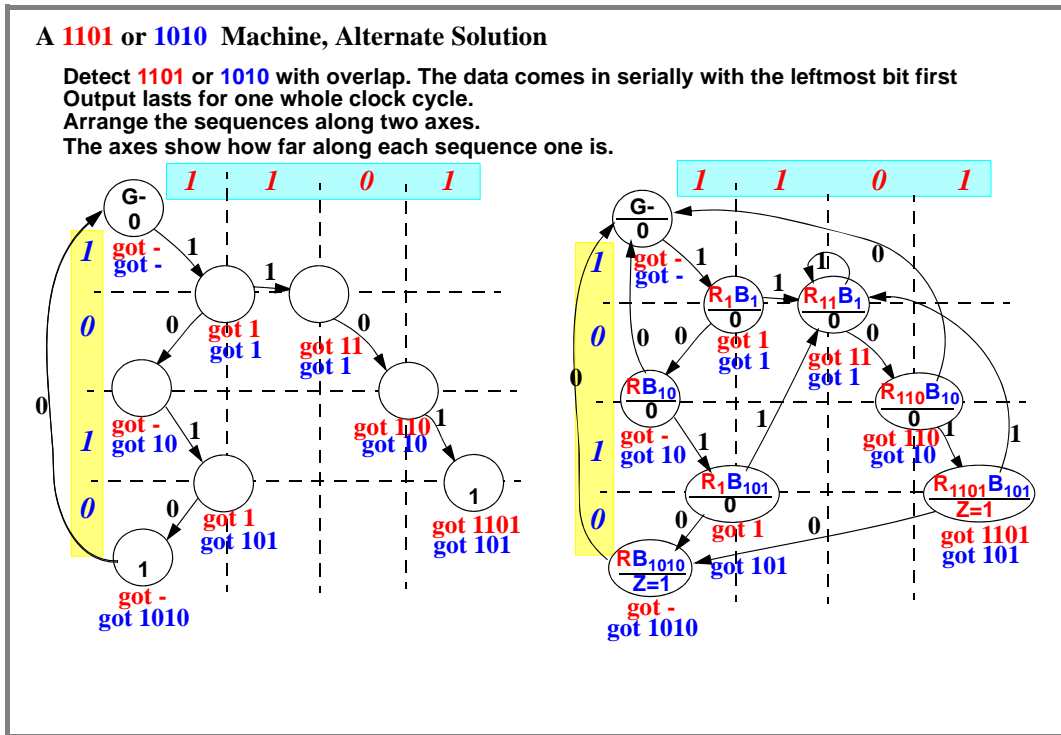
Do the **1010** graph vertically.

Do the **1101** graph horizontally.

Do the main sequence first

Fill in extra arrows

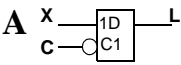
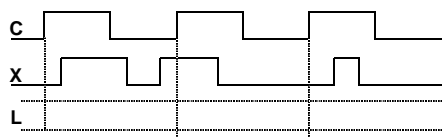


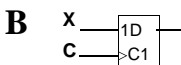
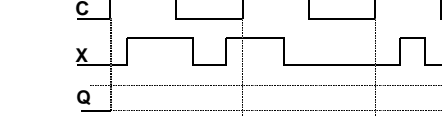


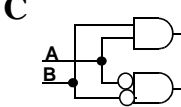
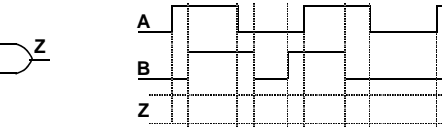


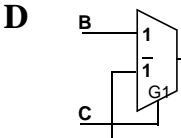
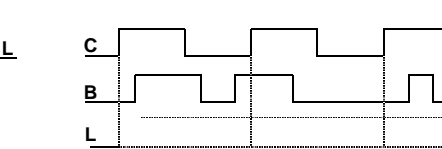
Latches etc from Dec '96.

Example.1-12 Sketch the Output Waveforms

A   *D Latch (inverted clock)
Transparent when?
Latched when?*

B   *D Flip-Flop
Rising edge triggered*

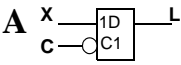
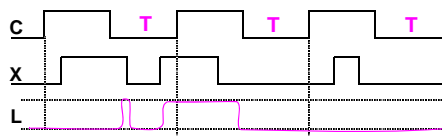
C   *Equivalent to
_____ gate*

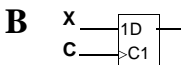
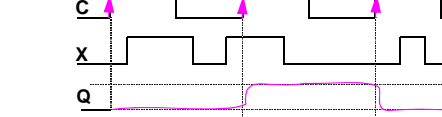
D   *D Latch
Transparent when C=1
Latched when C=0*

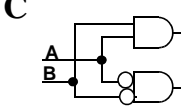
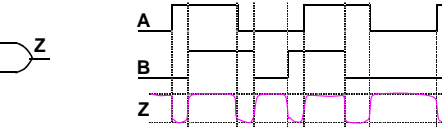


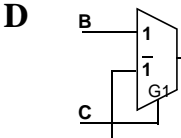
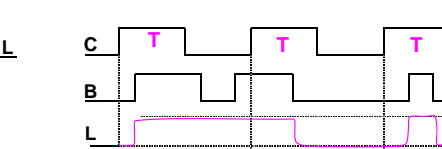
Latches etc from Dec '96.

Example.1-13 Sketch the Output Waveforms

A   *D Latch (inverted clock)
Transparent when clock low
Latched when high*

B   *D Flip-Flop
Rising edge triggered*

C   *XNOR gate*

D   *D Latch
Transparent when C=1
Latched when C=0*



Latches etc. from Dec '96.

Example.1-14 Sketch the Output Waveforms

E

Reset(L)-set(L) Latch
Set output =1 when RS=10
Set output to 0 when R=0
Store output when SR=11

F

Synchronous State Graph

State=Z

G

Synchronous State Graph

State A
State B

Show state as well as Z



Latches etc. from Dec '96.

Example.1-15 Sketch the Output Waveforms

E

Reset(L)-set(L) Latch
Set output =1 when RS=10
Set output to 0 when R=0
Store output when SR=11
Set dominant

F

Synchronous State Graph

Sketch the circuit.

State=Z

G

Synchronous State Graph

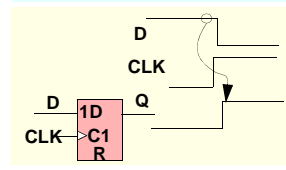
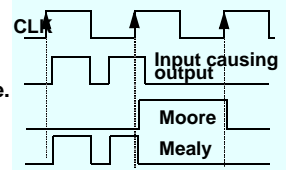
State A
State B

Show state as well as Z



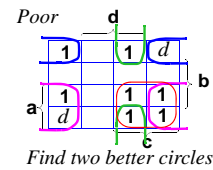
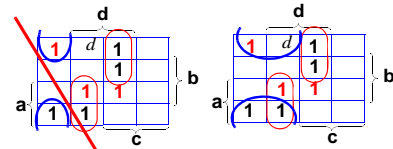
Last Words

- Read the question! All of it!
- Always check for $Ax + \bar{A} = A$ Use this to simplify anywhere
- Always reduce $A + \bar{A}x = A + x$ Use anywhere except for hazards
- Too many people say ~~$x + 1 = x$~~
- Do not confuse synchronous and asynchronous
 1. Synchronous
Not bothered by races. Assign states to reduce logic.
Not bothered by hazards. Don't try to cure them.
 2. Asynchronous
Assign states to be race free.
Must not have hazards.
- To tell Moore from Mealy in word problems
 1. Moore Outputs
Outputs will appear after the next active clock edge.
 2. Mealy Outputs
Outputs will appear after the input changes.
- Flip-flops
Sample D input just before the clock.
Transfer this to the output just after the clock.

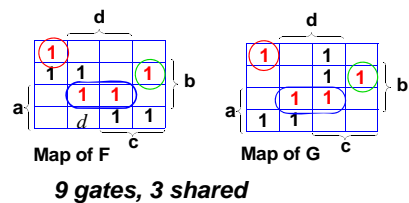
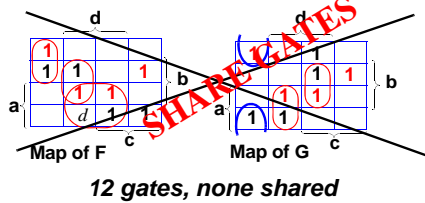


K-Maps; Common Errors

- Check Your Map Entries
One variable in the wrong square, you're toast!
- Check for Wraparound, and Wide Wrap Around
On logic-reduction K-maps
On K-maps used for state assignment
On State Tables used for race checking



- Don't Treat Multiple Output Problems Like Unrelated Circuits



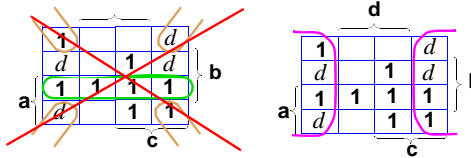
- A Karnaugh map may not give the simplest circuit, but it gives the simplest Σ of Π circuit.



K-Maps; Common Errors

- Do not confuse 5-variable maps and dual-output maps.
 - 5-variable use the largest circles,
 - Try to enlarge circle by using both maps.
- Multiple output
 - Sharing is very important to save gates
 - Often (esp exams) some smaller circles will give fewer gates.
 - sharing lowers gate count.
 - smaller circles increase letter count. } Balance these

- Multiple Output Problems
 - Do Half-Maps First (Except for PLAs)

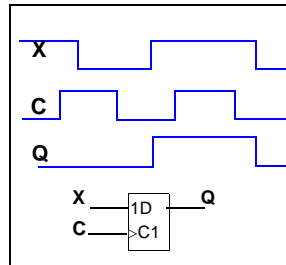
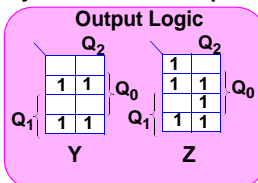
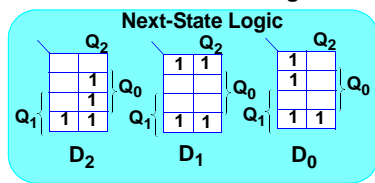


- No best friend, and other rules are heuristics, they help, but not always, and do not replace all thinking.



Common Errors When Designing Finite-State Machines

- Not rearranging the state table into Karnaugh map order.
- Not being able to choose Moore or Mealy outputs from a description.
 - Mealy output problems have words like:
 - "The output becomes true as soon as the final bit in the sequence arrives".
 - "The output does not have to wait for the clock."
 - "Choose an output type most likely to reduce the size of the circuit."
 - Moore output problems have words like:
 - "The output becomes true during the next state after the final input bit is received."
 - "The output must not transfer any glitches that are on the input."
- Changing states at some other place than the active clock edge. Often erroneously done when working with timing diagrams
- Forgetting that FSMs are multiple output machines. Watch for shared gates when you circle the maps.



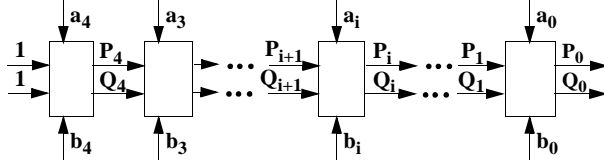


Lab Related Problem (Not Covered 2007)

The 5-bit iterative comparator shown compares two 5-bit binary numbers $a_4a_3a_2a_1a_0$ and $b_4b_3b_2b_1b_0$.

Fill in each square of the K-map for the output Q_i of block i , with 1, 0, or d as appropriate.

$P_i = 1$ if $a_4 \dots a_i \geq b_4 \dots b_i$
 $Q_i = 1$ if $b_4 \dots b_i \geq a_4 \dots a_i$



$P_i = 1$ if $a_4 \dots a_i \geq b_4 \dots b_i$
 $Q_i = 1$ if $b_4 \dots b_i \geq a_4 \dots a_i$

$P_4 = 1$ if $a_4 \geq b_4$
 $Q_4 = 1$ if $b_4 \geq a_4$

P and Q cannot both be zero since one number must be greater than or equal

$P_3 = 1$ if $(a_3 \geq b_3) \& (P_4 = Q_4 = 1)$ or $P_4 > Q_4$
 $Q_3 = 1$ if $(b_3 \geq a_3) \& (P_4 = Q_4 = 1)$ or $Q_4 > P_4$

$P_i = 1$ if $(a_i + \bar{b}_i) \& (P_{i-1}Q_{i-1})$ or $P_{i-1}\bar{Q}_{i-1}$

$Q_i = (b_i + \bar{a}_i)P_{i-1}Q_{i-1} + \bar{P}_{i-1}Q_{i-1}$
 with "d"s reduces to $(b_i + \bar{a}_i)Q_{i-1} + \bar{P}_{i-1}$

	$P_{i+1}Q_{i+1}$			
$a_i b_i$	00	01	11	10
00				
01				
11				
10				

Map of Q_i

The proper use of d is important.
 Hint: One column is all ds.

seven "1"s and four "d"s

