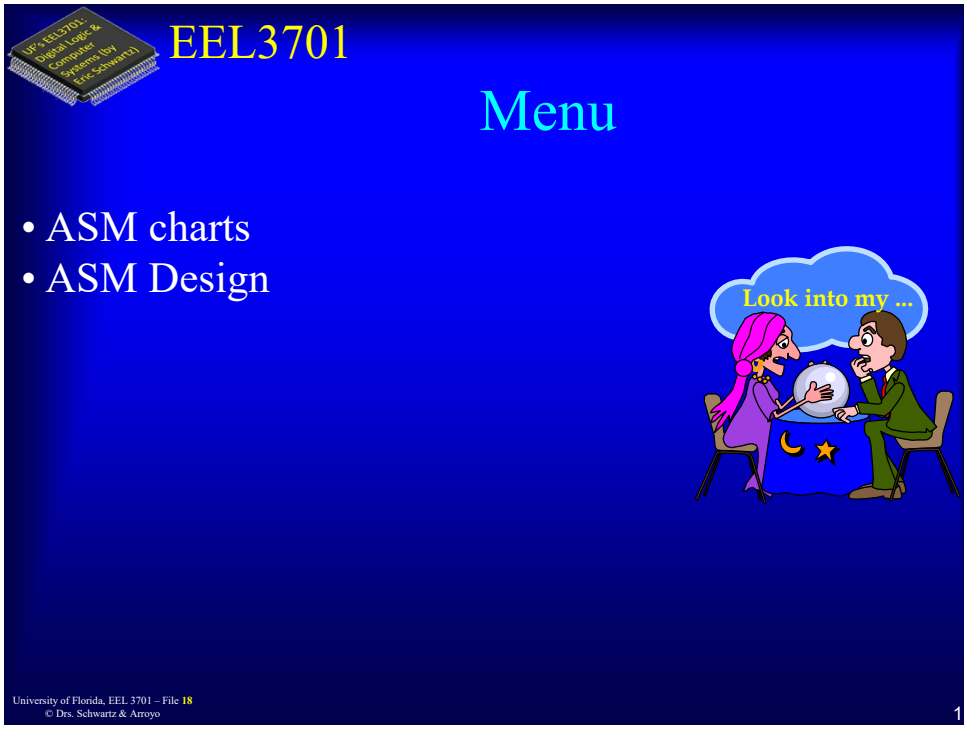
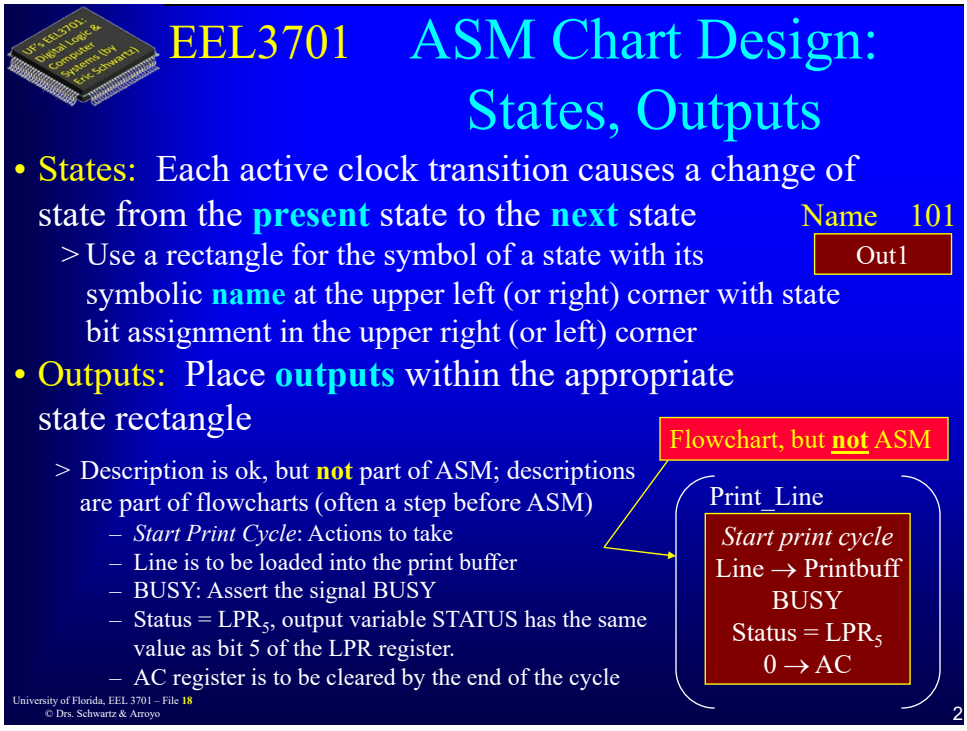
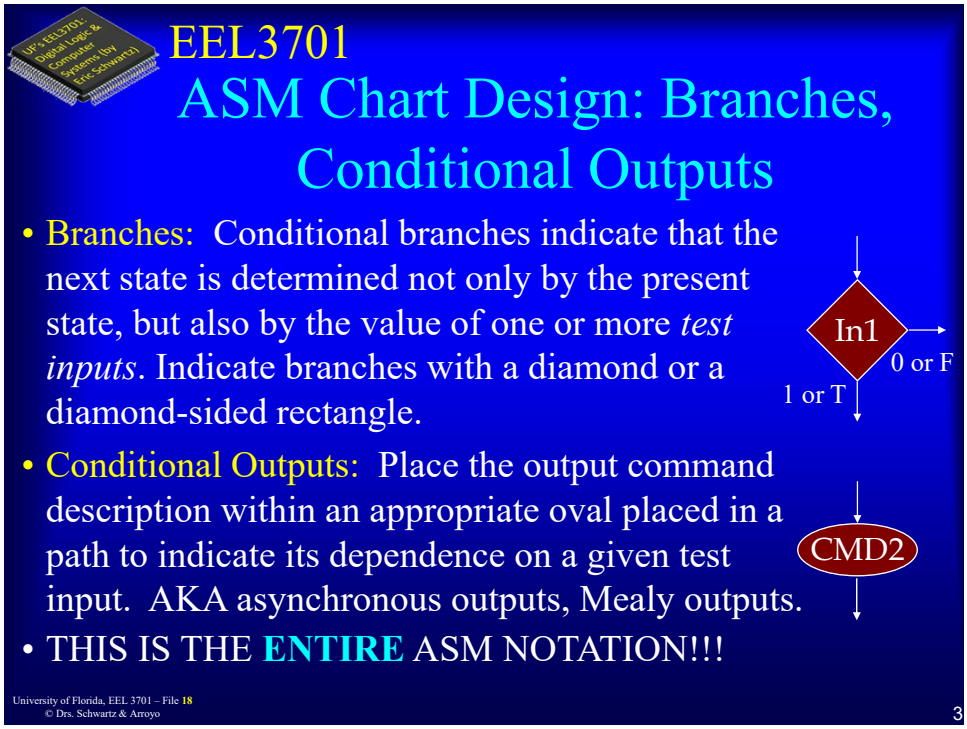

EEL3701
Menu
• ASM charts
• ASM Design
Look into my ...
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
1

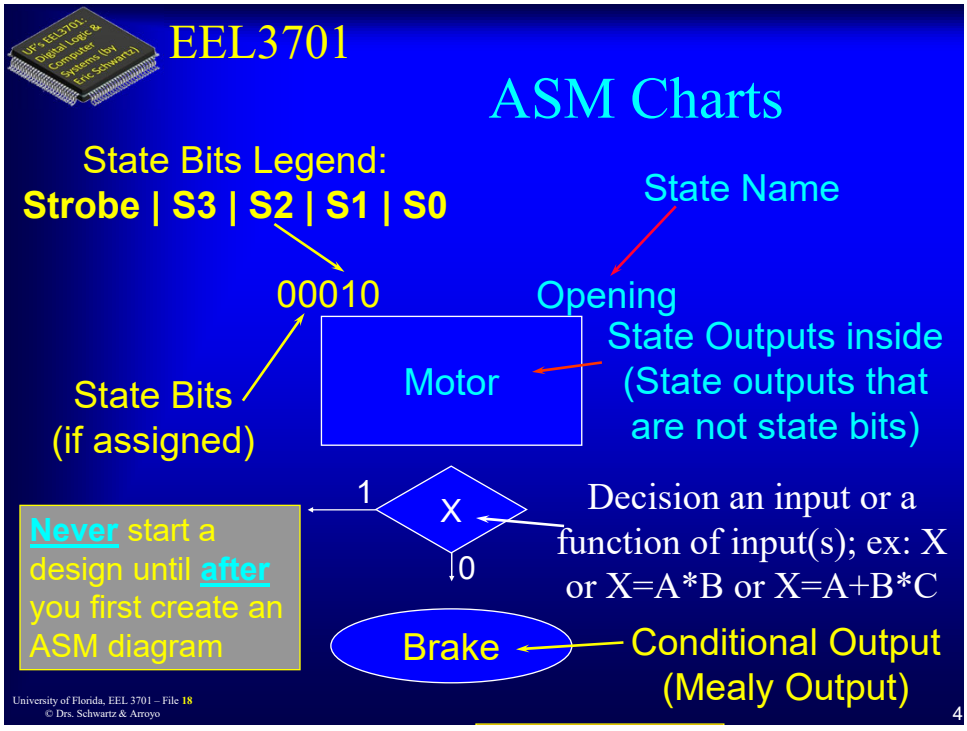
1


EEL3701
ASM Chart Design: States, Outputs
• States: Each active clock transition causes a change of state from the present state to the next state
> Use a rectangle for the symbol of a state with its symbolic name at the upper left (or right) corner with state bit assignment in the upper right (or left) corner
Name 101
Out1
• Outputs: Place outputs within the appropriate state rectangle
> Description is ok, but not part of ASM; descriptions are part of flowcharts (often a step before ASM)
– Start Print Cycle: Actions to take
– Line is to be loaded into the print buffer
– BUSY: Assert the signal BUSY
– Status = LPR5, output variable STATUS has the same value as bit 5 of the LPR register.
– AC register is to be cleared by the end of the cycle
Flowchart, but not ASM
Print_Line
Start print cycle
Line → Printbuff
BUSY
Status = LPR5
0 → AC
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
2

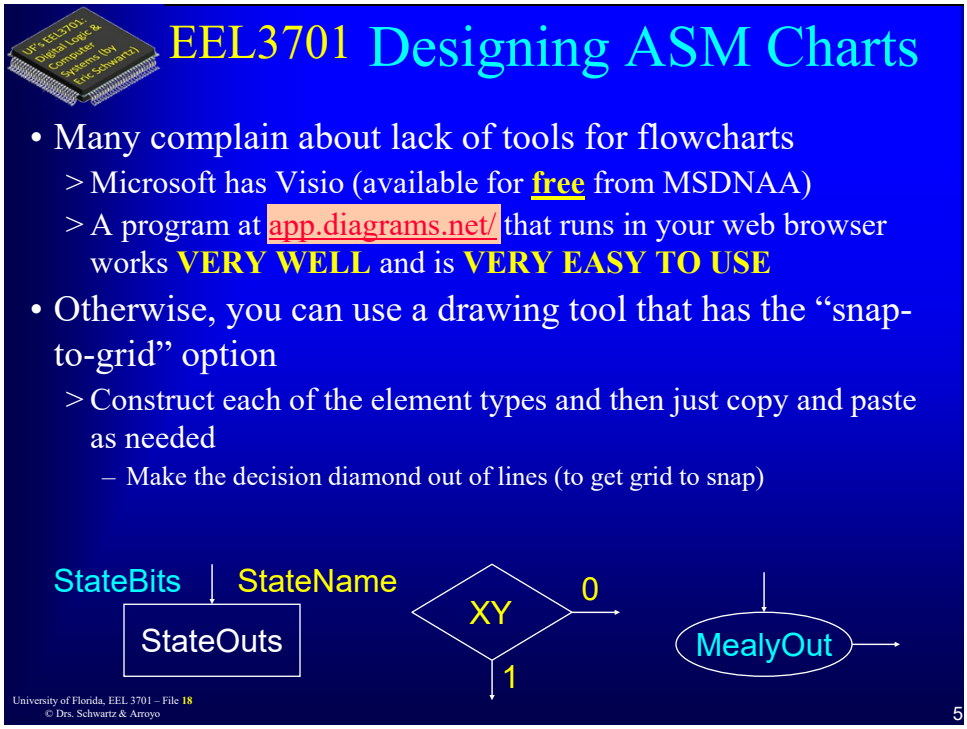
2


EEL3701
ASM Chart Design: Branches, Conditional Outputs
• Branches: Conditional branches indicate that the next state is determined not only by the present state, but also by the value of one or more *test inputs*. Indicate branches with a diamond or a diamond-sided rectangle.
• Conditional Outputs: Place the output command description within an appropriate oval placed in a path to indicate its dependence on a given test input. AKA asynchronous outputs, Mealy outputs.
• THIS IS THE ENTIRE ASM NOTATION!!!
In1
0 or F
1 or T
CMD2
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
3

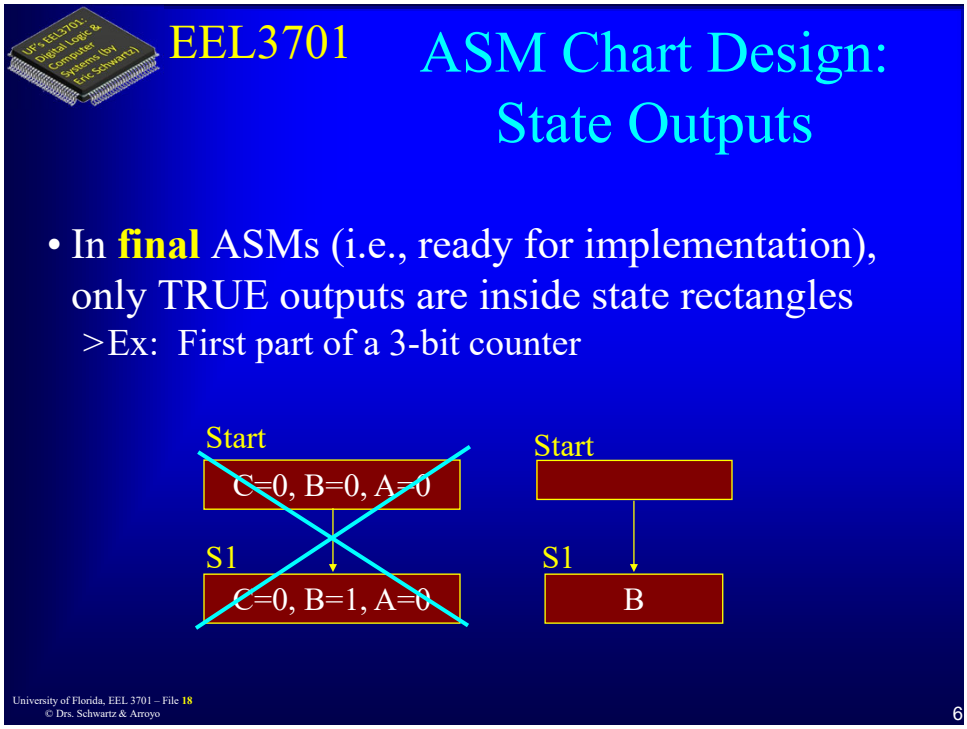
3


EEL3701
ASM Charts
State Bits Legend:
Strobe | S3 | S2 | S1 | S0
State Name
00010
Opening
State Bits
(if assigned)
Motor
State Outputs inside
(State outputs that
are not state bits)
1
X
0
Never start a design until after you first create an ASM diagram
Decision an input or a function of input(s); ex: X or X=A*B or X=A+B*C
Brake
Conditional Output
(Mealy Output)
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
4

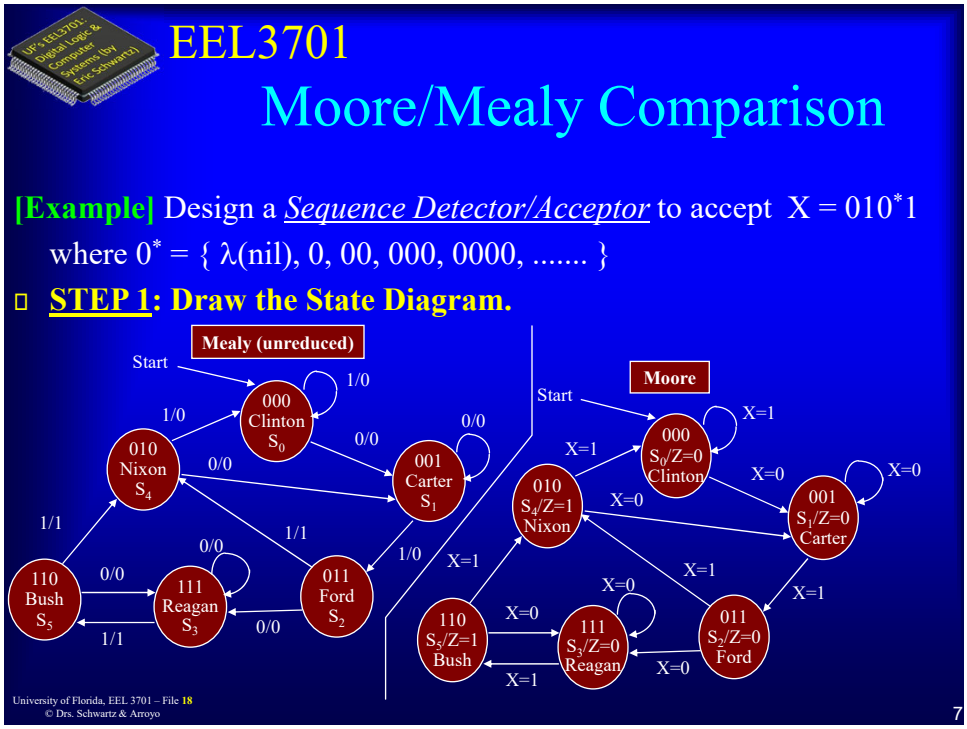
4


EEL3701 Designing ASM Charts
• Many complain about lack of tools for flowcharts
> Microsoft has Visio (available for free from MSDNAA)
> A program at app.diagrams.net/ that runs in your web browser works VERY WELL and is VERY EASY TO USE
• Otherwise, you can use a drawing tool that has the "snap-to-grid" option
> Construct each of the element types and then just copy and paste as needed
– Make the decision diamond out of lines (to get grid to snap)
StateBits
StateName
StateOuts
XY
0
1
MealyOut
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
5

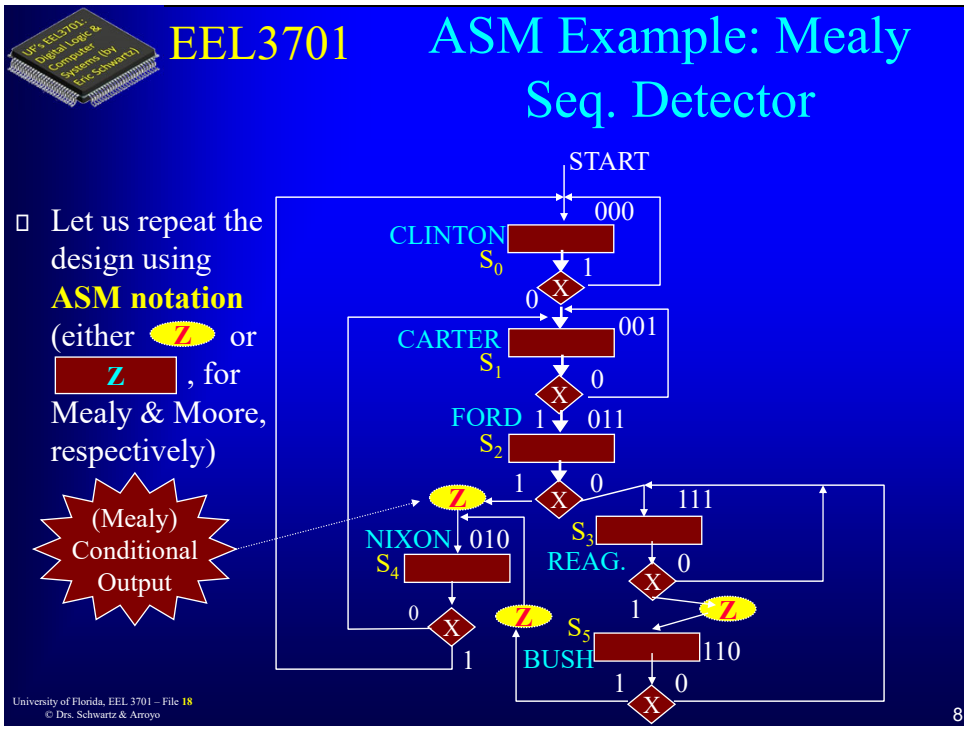
5


EEL3701
ASM Chart Design: State Outputs
• In final ASMs (i.e., ready for implementation), only TRUE outputs are inside state rectangles
>Ex: First part of a 3-bit counter
Start
C=0, B=0, A=0
S1
C=0, B=1, A=0
Start
S1
B
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
6

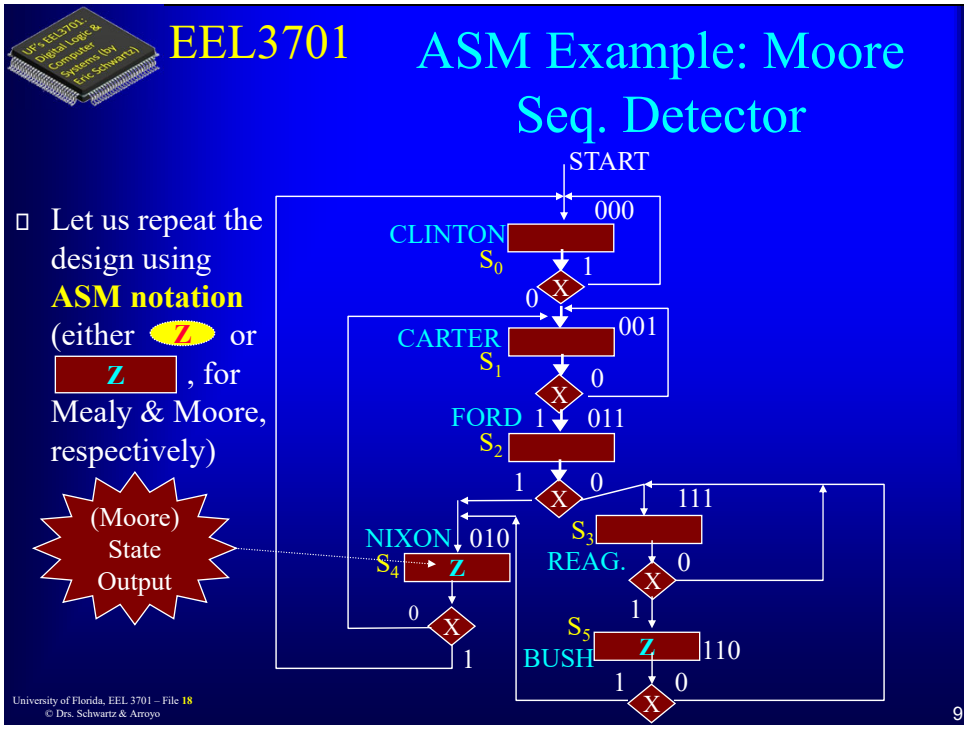
6


EEL3701
Moore/Mealy Comparison
[Example] Design a Sequence Detector/Acceptor to accept X = 010*1
where 0* = { λ(nil), 0, 00, 000, 0000, ... }
STEP 1: Draw the State Diagram.
Mealy (unreduced)
Start
000 Clinton S0
010 Nixon S4
001 Carter S1
110 Bush S5
111 Reagan S3
011 Ford S2
1/0
0/0
1/1
Moore
000 S0/Z=0 Clinton
010 S4/Z=1 Nixon
001 S1/Z=0 Carter
110 S5/Z=1 Bush
111 S3/Z=0 Reagan
011 S2/Z=0 Ford
X=1
X=0
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
7

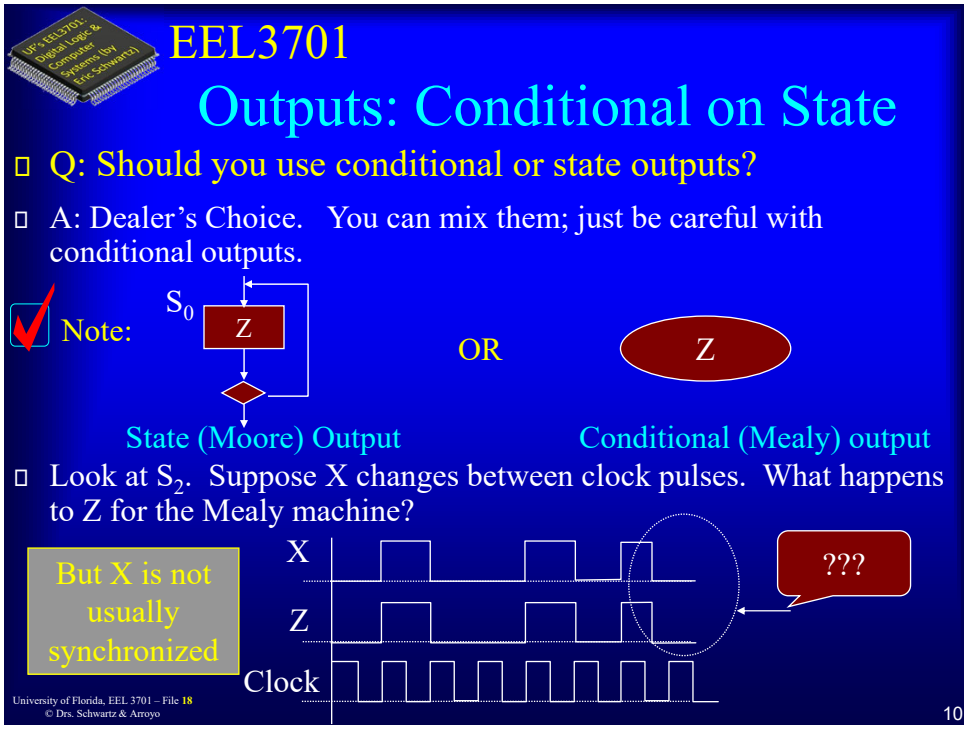
7


EEL3701
ASM Example: Mealy Seq. Detector
Let us repeat the design using ASM notation (either Z or Z, for Mealy & Moore, respectively)
(Mealy) Conditional Output
START
CLINTON S0
000
CARTER S1
001
FORD S2
011
NIXON S4
010
REAG. S3
111
BUSH S5
110
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
8

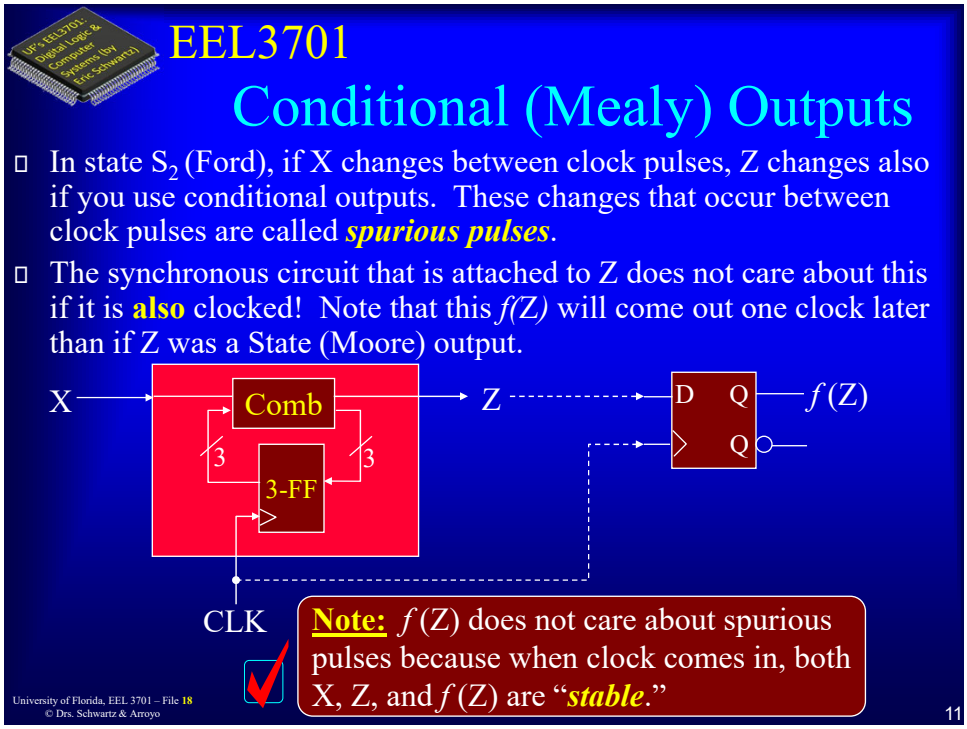
8


EEL3701
ASM Example: Moore Seq. Detector
Let us repeat the design using ASM notation (either Z or Z, for Mealy & Moore, respectively)
(Moore) State Output
START
CLINTON
S_0
000
1
0
X
CARTER
S_1
001
0
FORD
S_2
011
1
0
111
S_3
REAG.
NIXON
S_4
010
Z
S_5
BUSH
110
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
9

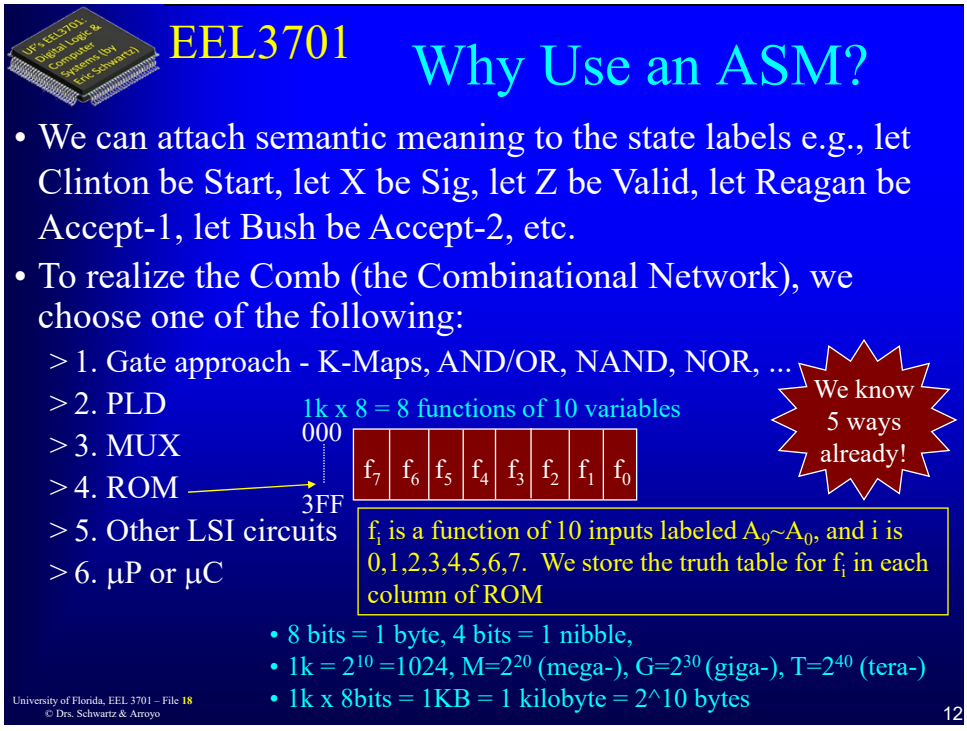
9


EEL3701
Outputs: Conditional on State
Q: Should you use conditional or state outputs?
A: Dealer's Choice. You can mix them; just be careful with conditional outputs.
Note:
S_0
Z
OR
Z
State (Moore) Output
Conditional (Mealy) output
Look at S_2. Suppose X changes between clock pulses. What happens to Z for the Mealy machine?
But X is not usually synchronized
X
Z
Clock
???
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
10

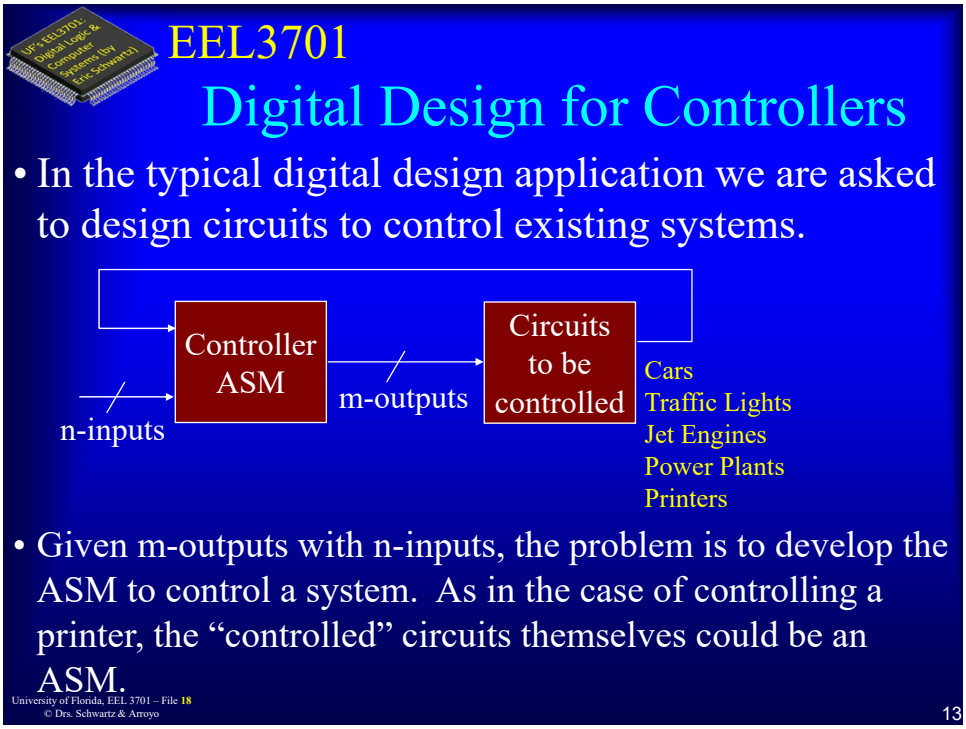
10


EEL3701
Conditional (Mealy) Outputs
In state S_2 (Ford), if X changes between clock pulses, Z changes also if you use conditional outputs. These changes that occur between clock pulses are called ***spurious pulses***.
The synchronous circuit that is attached to Z does not care about this if it is **also** clocked! Note that this *f(Z)* will come out one clock later than if Z was a State (Moore) output.
X
Comb
Z
D
Q
Q
f (Z)
3
3
3-FF
CLK
Note: *f* (Z) does not care about spurious pulses because when clock comes in, both X, Z, and *f* (Z) are "***stable***."
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
11

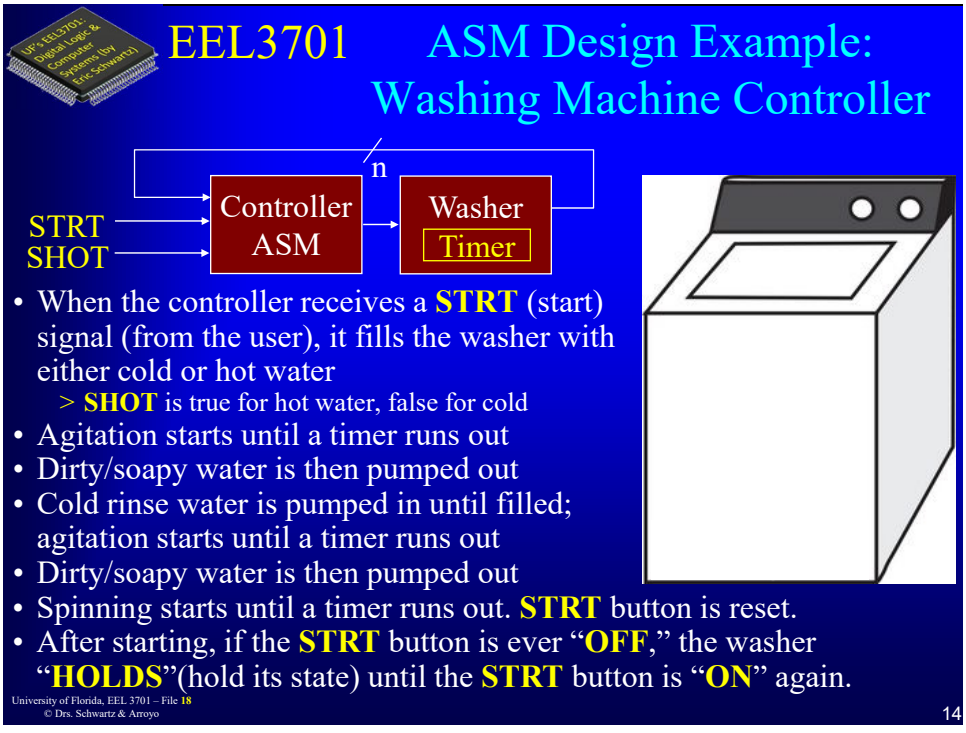
11


EEL3701
Why Use an ASM?
We can attach semantic meaning to the state labels e.g., let Clinton be Start, let X be Sig, let Z be Valid, let Reagan be Accept-1, let Bush be Accept-2, etc.
To realize the Comb (the Combinational Network), we choose one of the following:
> 1. Gate approach - K-Maps, AND/OR, NAND, NOR, ...
> 2. PLD
> 3. MUX
> 4. ROM
> 5. Other LSI circuits
> 6. µP or µC
1k x 8 = 8 functions of 10 variables
000
3FF
f_7 f_6 f_5 f_4 f_3 f_2 f_1 f_0
We know 5 ways already!
f_i is a function of 10 inputs labeled A_9~A_0, and i is 0,1,2,3,4,5,6,7. We store the truth table for f_i in each column of ROM
• 8 bits = 1 byte, 4 bits = 1 nibble,
• 1k = 2^{10} =1024, M=2^{20} (mega-), G=2^{30} (giga-), T=2^{40} (tera-)
• 1k x 8bits = 1KB = 1 kilobyte = 2^10 bytes
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
12

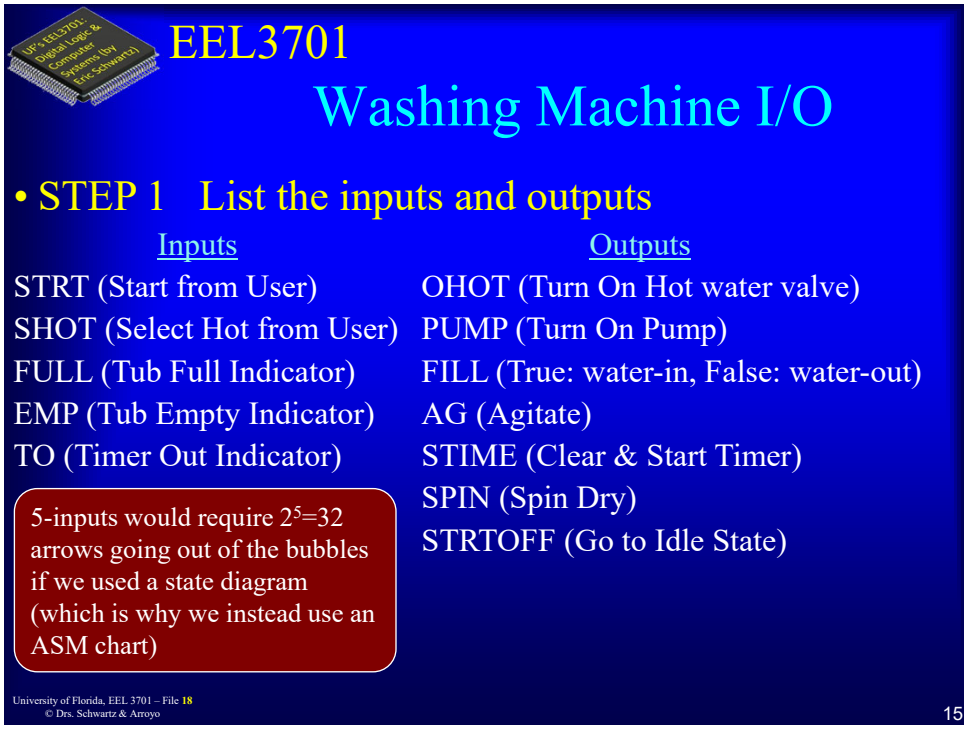
12


EEL3701
Digital Design for Controllers
• In the typical digital design application we are asked to design circuits to control existing systems.
Controller ASM
n-inputs
m-outputs
Circuits to be controlled
Cars
Traffic Lights
Jet Engines
Power Plants
Printers
• Given m-outputs with n-inputs, the problem is to develop the ASM to control a system. As in the case of controlling a printer, the "controlled" circuits themselves could be an ASM.
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
13

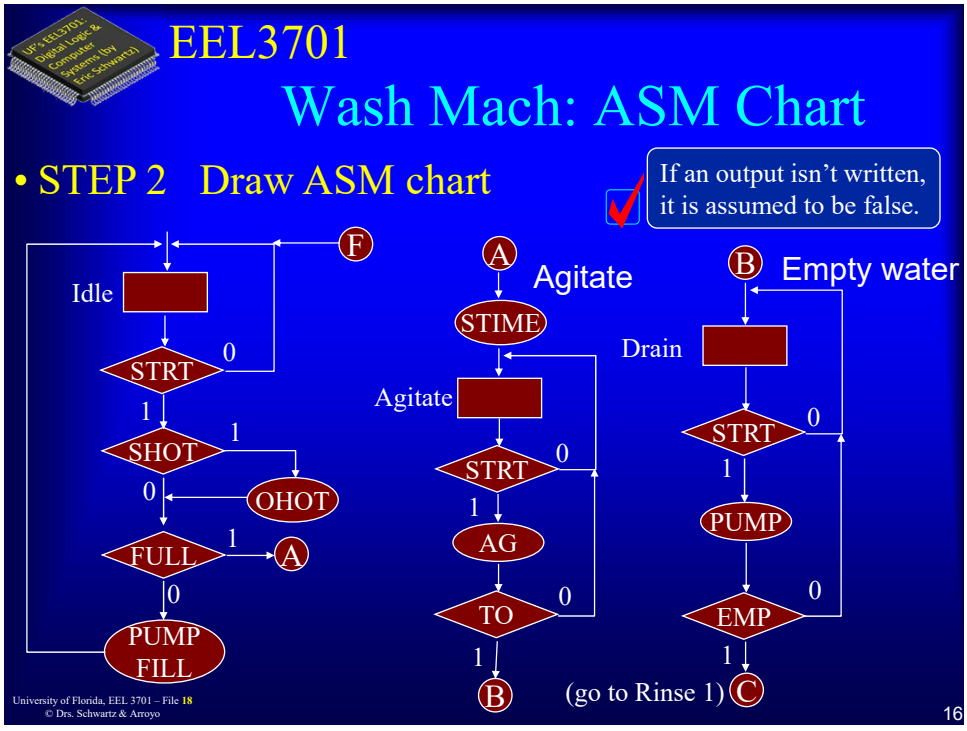
13


EEL3701
ASM Design Example:
Washing Machine Controller
n
STRT
SHOT
Controller ASM
Washer
Timer
• When the controller receives a STRT (start) signal (from the user), it fills the washer with either cold or hot water
> SHOT is true for hot water, false for cold
• Agitation starts until a timer runs out
• Dirty/soapy water is then pumped out
• Cold rinse water is pumped in until filled; agitation starts until a timer runs out
• Dirty/soapy water is then pumped out
• Spinning starts until a timer runs out. STRT button is reset.
• After starting, if the STRT button is ever "OFF," the washer "HOLDS"(hold its state) until the STRT button is "ON" again.
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
14

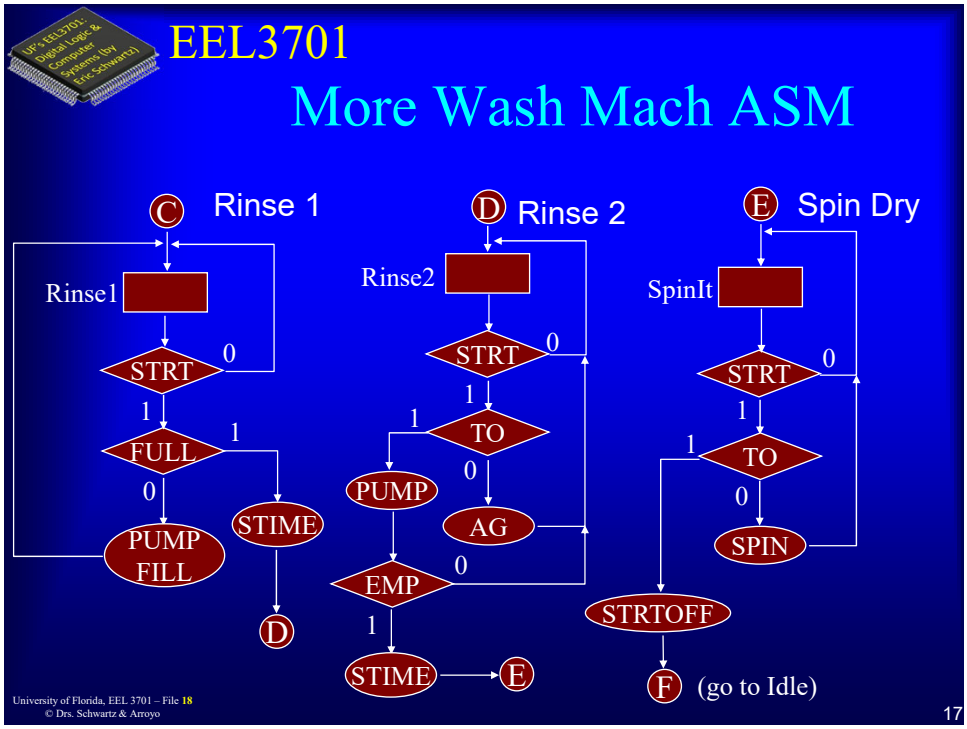
14


EEL3701
Washing Machine I/O
• STEP 1 List the inputs and outputs
Inputs
STRT (Start from User)
SHOT (Select Hot from User)
FULL (Tub Full Indicator)
EMP (Tub Empty Indicator)
TO (Timer Out Indicator)
5-inputs would require 2^5=32 arrows going out of the bubbles if we used a state diagram (which is why we instead use an ASM chart)
Outputs
OHOT (Turn On Hot water valve)
PUMP (Turn On Pump)
FILL (True: water-in, False: water-out)
AG (Agitate)
STIME (Clear & Start Timer)
SPIN (Spin Dry)
STRTOFF (Go to Idle State)
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
15

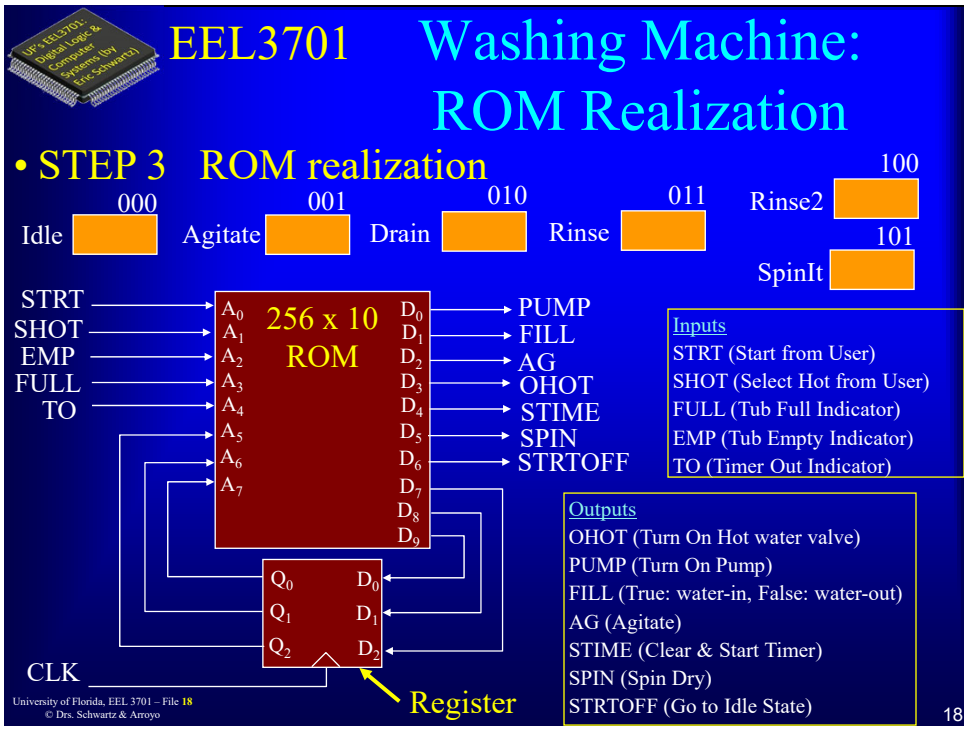
15


EEL3701
Wash Mach: ASM Chart
• STEP 2 Draw ASM chart
If an output isn't written, it is assumed to be false.
Idle
STRT
SHOT
OHOT
FULL
PUMP
FILL
F
A
Agitate
STIME
Agitate
STRT
AG
TO
B
Empty water
Drain
STRT
PUMP
EMP
(go to Rinse 1)
C
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
16

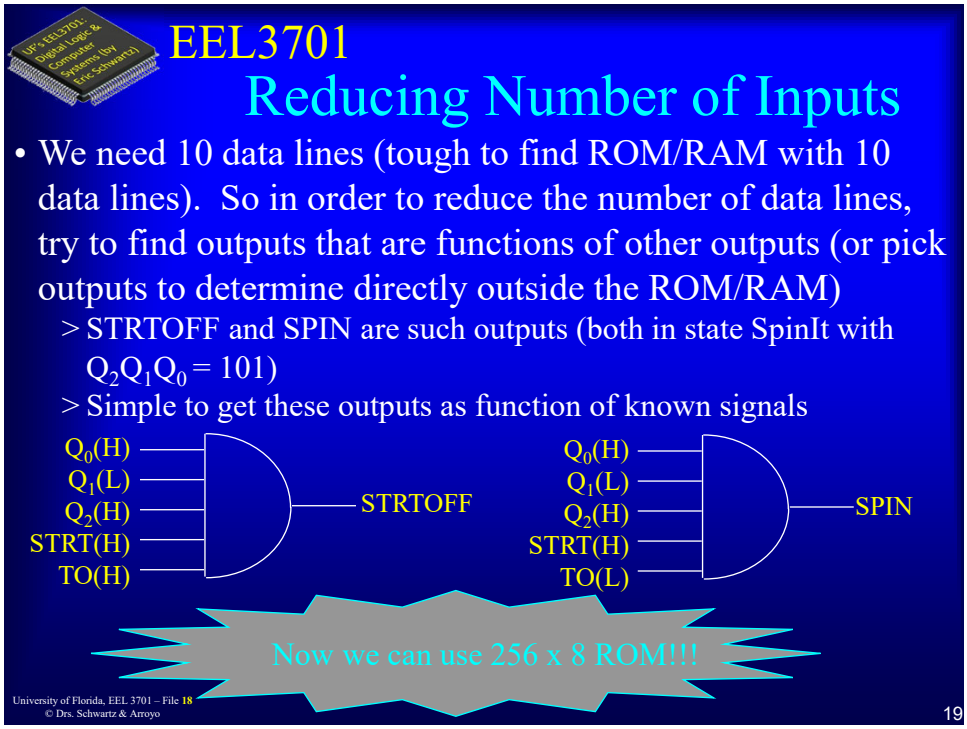
16


EEL3701
More Wash Mach ASM
C Rinse 1
Rinse1
STRT
0
1
FULL
1
0
PUMP
FILL
STIME
D
D Rinse 2
Rinse2
STRT
0
1
TO
1
0
PUMP
AG
EMP
0
1
STIME
E
E Spin Dry
SpinIt
STRT
0
1
TO
1
0
SPIN
STRTOFF
F (go to Idle)
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
17

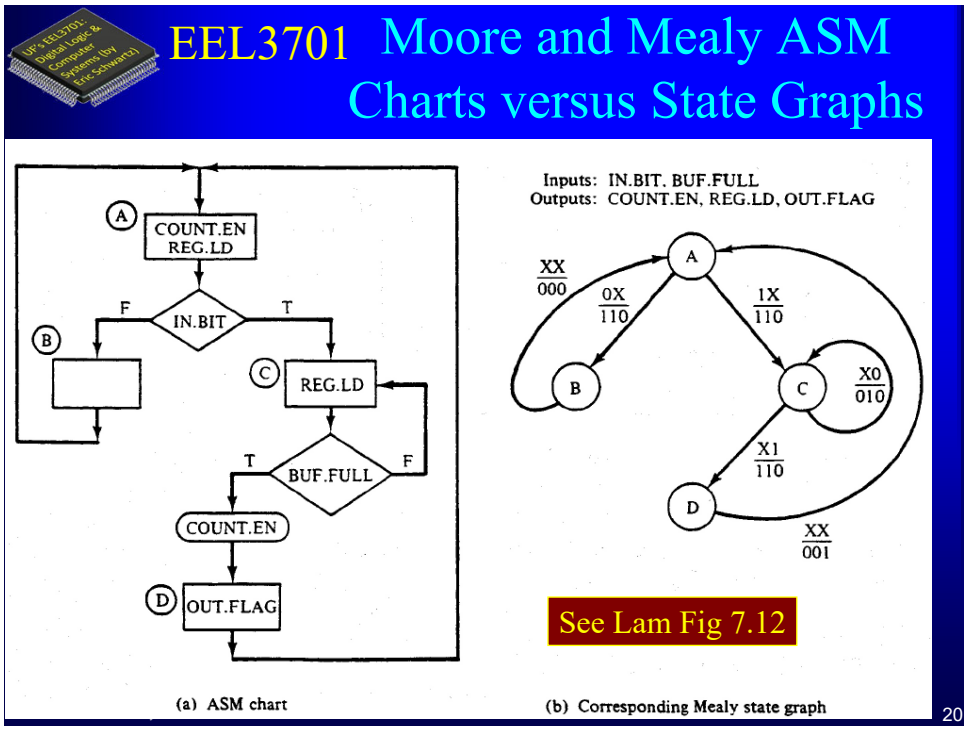
17


EEL3701
Washing Machine:
ROM Realization
• STEP 3 ROM realization
000
Idle
001
Agitate
010
Drain
011
Rinse
100
Rinse2
101
SpinIt
STRT
SHOT
EMP
FULL
TO
A0 A1 A2 A3 A4 A5 A6 A7
256 x 10
ROM
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9
PUMP
FILL
AG
OHOT
STIME
SPIN
STRTOFF
Q0 Q1 Q2
D0 D1 D2
CLK
Register
Inputs
STRT (Start from User)
SHOT (Select Hot from User)
FULL (Tub Full Indicator)
EMP (Tub Empty Indicator)
TO (Timer Out Indicator)
Outputs
OHOT (Turn On Hot water valve)
PUMP (Turn On Pump)
FILL (True: water-in, False: water-out)
AG (Agitate)
STIME (Clear & Start Timer)
SPIN (Spin Dry)
STRTOFF (Go to Idle State)
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
18

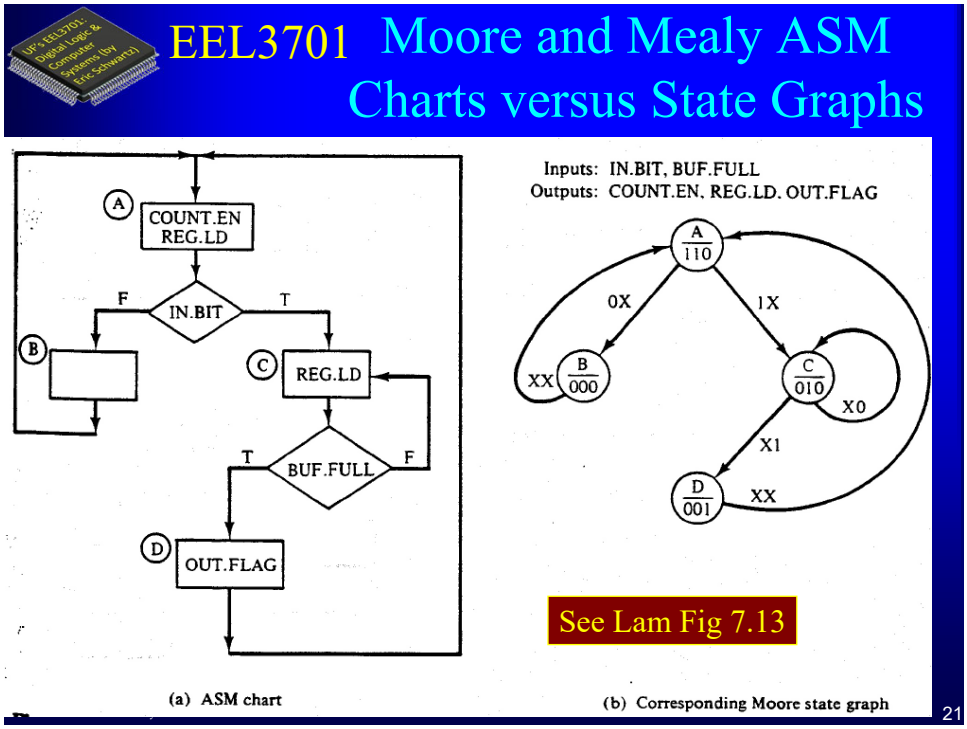
18


EEL3701
Reducing Number of Inputs
• We need 10 data lines (tough to find ROM/RAM with 10 data lines). So in order to reduce the number of data lines, try to find outputs that are functions of other outputs (or pick outputs to determine directly outside the ROM/RAM)
> STRTOFF and SPIN are such outputs (both in state SpinIt with $Q_2Q_1Q_0 = 101$)
> Simple to get these outputs as function of known signals
Q_0(H)
Q_1(L)
Q_2(H)
STRT(H)
TO(H)
STRTOFF
Q_0(H)
Q_1(L)
Q_2(H)
STRT(H)
TO(L)
SPIN
Now we can use 256 x 8 ROM!!!
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
19

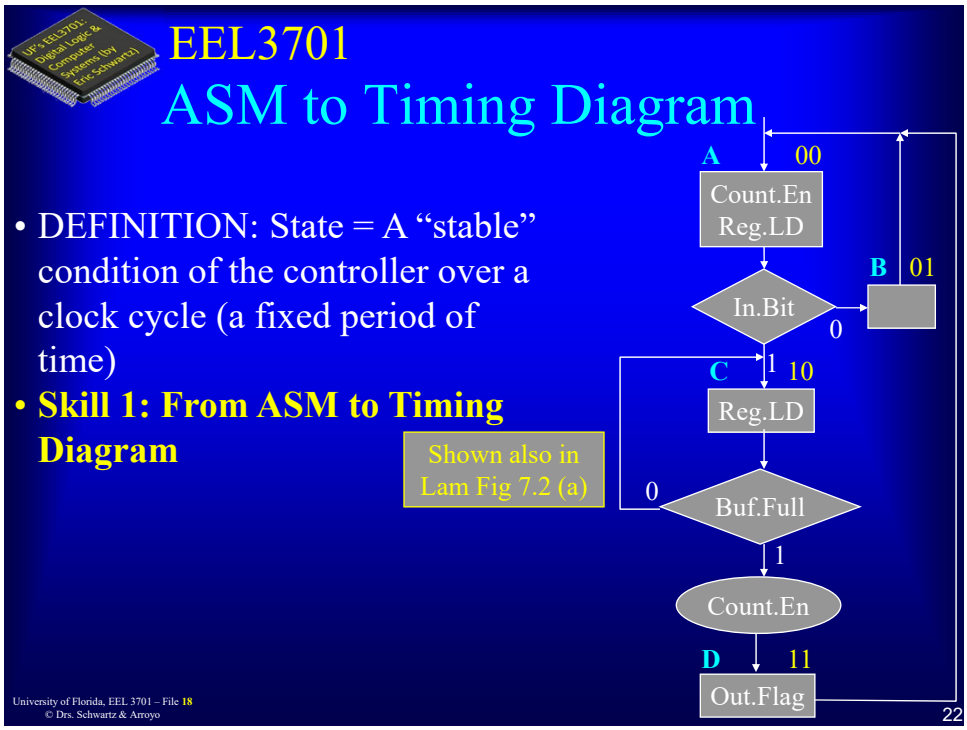
19


EEL3701 Moore and Mealy ASM Charts versus State Graphs
A
COUNT.EN
REG.LD
F
IN.BIT
T
B
C
REG.LD
T
BUF.FULL
F
COUNT.EN
D
OUT.FLAG
(a) ASM chart
Inputs: IN.BIT, BUF.FULL
Outputs: COUNT.EN, REG.LD, OUT.FLAG
XX
000
0X
110
1X
110
X0
010
X1
110
XX
001
See Lam Fig 7.12
(b) Corresponding Mealy state graph
20

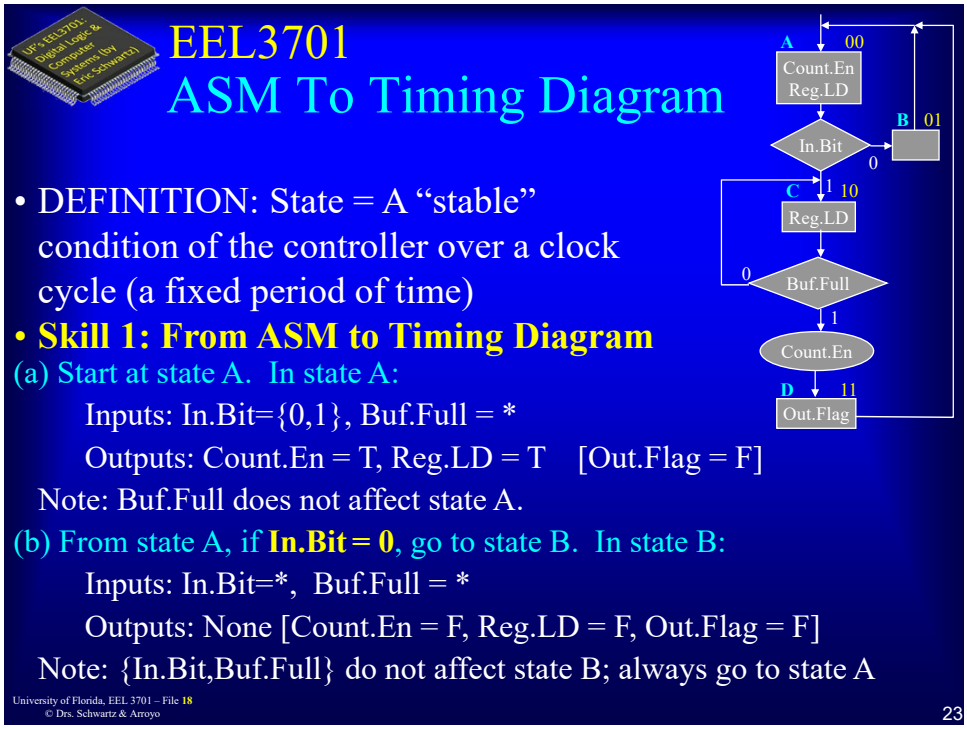
20


EEL3701 Moore and Mealy ASM Charts versus State Graphs
A
COUNT.EN
REG.LD
F
IN.BIT
T
B
C
REG.LD
T
BUF.FULL
F
D
OUT.FLAG
(a) ASM chart
Inputs: IN.BIT, BUF.FULL
Outputs: COUNT.EN, REG.LD, OUT.FLAG
A
110
0X
1X
XX
B
000
C
010
X0
X1
D
001
XX
See Lam Fig 7.13
(b) Corresponding Moore state graph
21

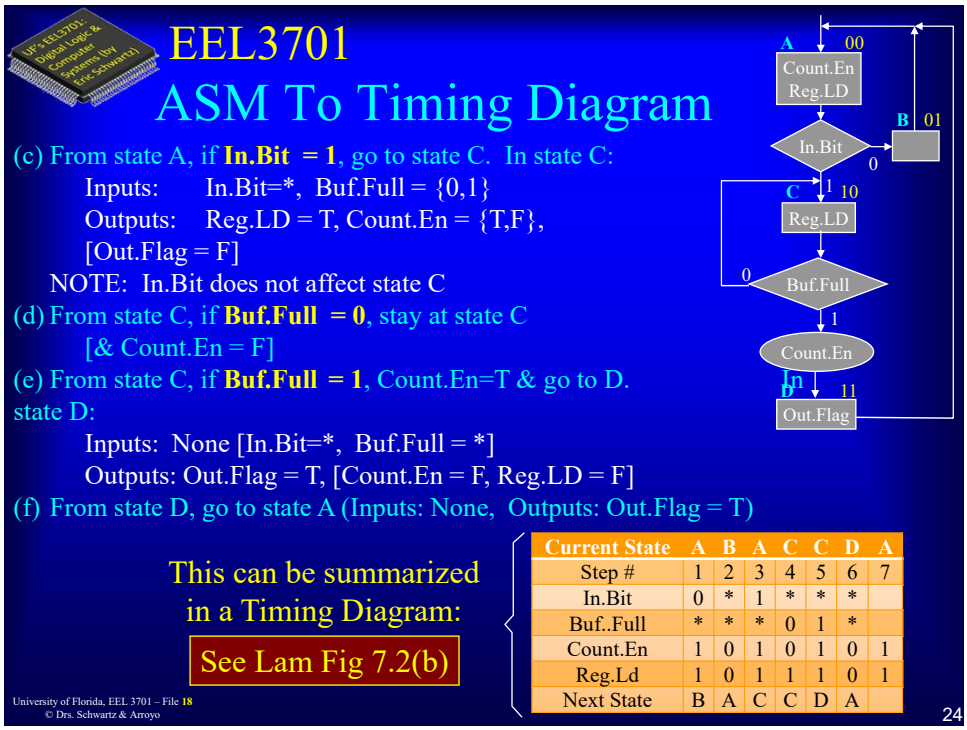
21


EEL3701
ASM to Timing Diagram
• DEFINITION: State = A "stable" condition of the controller over a clock cycle (a fixed period of time)
• Skill 1: From ASM to Timing Diagram
Shown also in Lam Fig 7.2 (a)
A
00
Count.En
Reg.LD
B
01
In.Bit
0
1
C
10
Reg.LD
0
Buf.Full
1
Count.En
D
11
Out.Flag
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
22

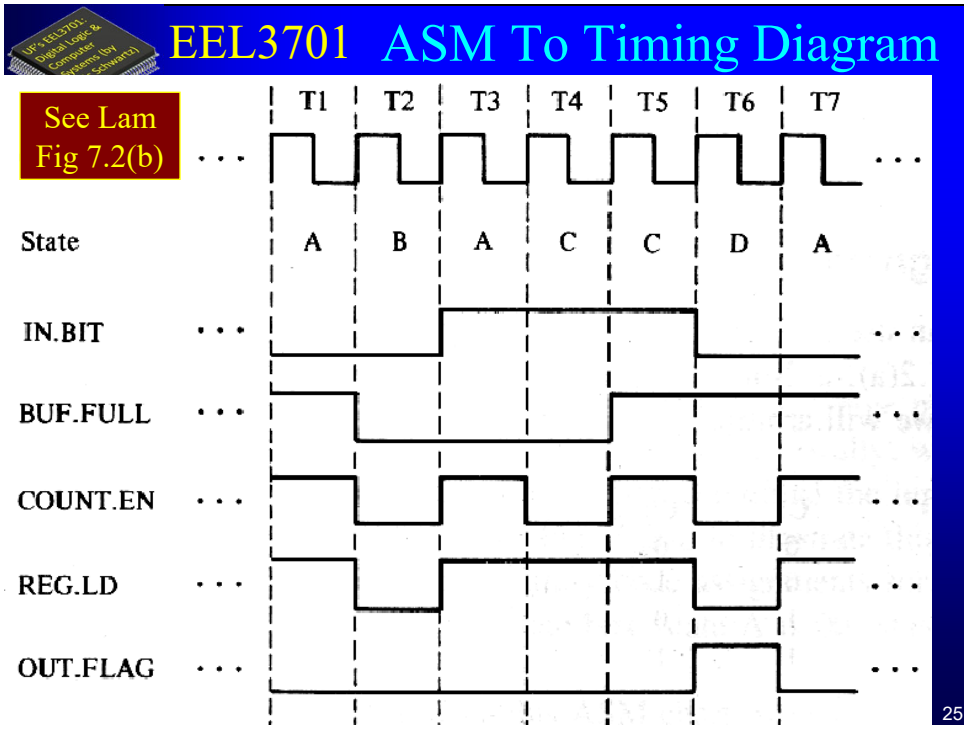
22


EEL3701
ASM To Timing Diagram
• DEFINITION: State = A "stable" condition of the controller over a clock cycle (a fixed period of time)
• Skill 1: From ASM to Timing Diagram
(a) Start at state A. In state A:
Inputs: In.Bit={0,1}, Buf.Full = *
Outputs: Count.En = T, Reg.LD = T [Out.Flag = F]
Note: Buf.Full does not affect state A.
(b) From state A, if In.Bit = 0, go to state B. In state B:
Inputs: In.Bit=*, Buf.Full = *
Outputs: None [Count.En = F, Reg.LD = F, Out.Flag = F]
Note: {In.Bit,Buf.Full} do not affect state B; always go to state A
A 00
Count.En
Reg.LD
In.Bit
B 01
C 10
Reg.LD
Buf.Full
Count.En
D 11
Out.Flag
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
23

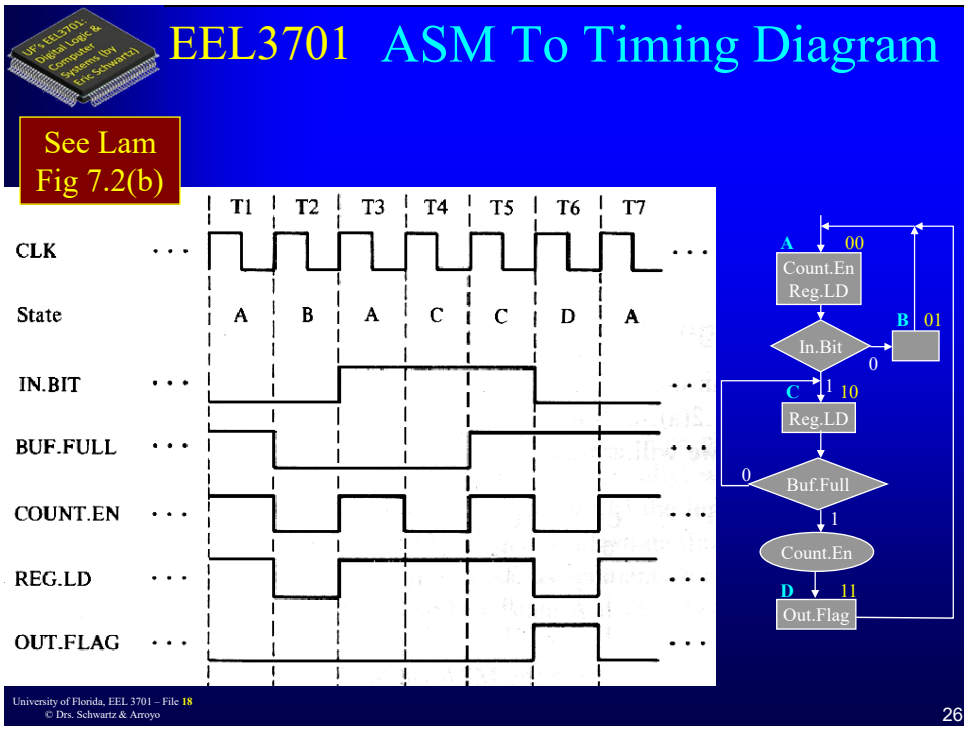
23


EEL3701
ASM To Timing Diagram
(c) From state A, if In.Bit = 1, go to state C. In state C:
Inputs: In.Bit=*, Buf.Full = {0,1}
Outputs: Reg.LD = T, Count.En = {T,F},
[Out.Flag = F]
NOTE: In.Bit does not affect state C
(d) From state C, if Buf.Full = 0, stay at state C
[& Count.En = F]
(e) From state C, if Buf.Full = 1, Count.En=T & go to D.
state D:
Inputs: None [In.Bit=*, Buf.Full = *]
Outputs: Out.Flag = T, [Count.En = F, Reg.LD = F]
(f) From state D, go to state A (Inputs: None, Outputs: Out.Flag = T)
This can be summarized in a Timing Diagram:
See Lam Fig 7.2(b)
Current State A B A C C D A
Step # 1 2 3 4 5 6 7
In.Bit 0 * 1 * * *
Buf..Full * * * 0 1 *
Count.En 1 0 1 0 1 0 1
Reg.Ld 1 0 1 1 1 0 1
Next State B A C C D A
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
24

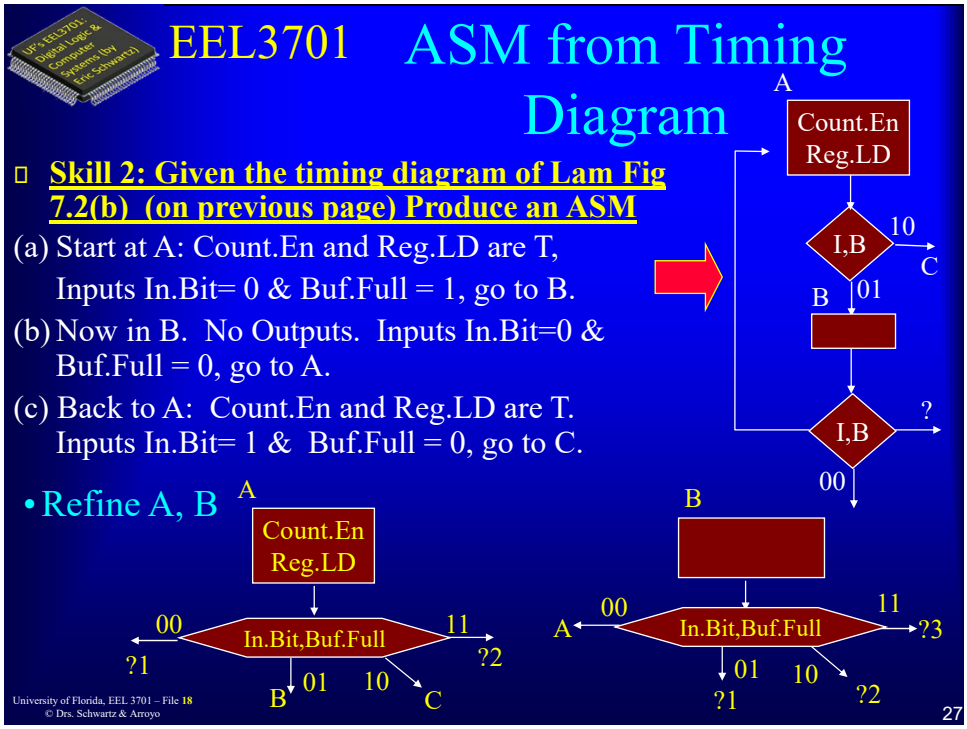
24


EEL3701 ASM To Timing Diagram
See Lam Fig 7.2(b)
T1 T2 T3 T4 T5 T6 T7
State
A B A C C D A
IN.BIT
BUF.FULL
COUNT.EN
REG.LD
OUT.FLAG
25

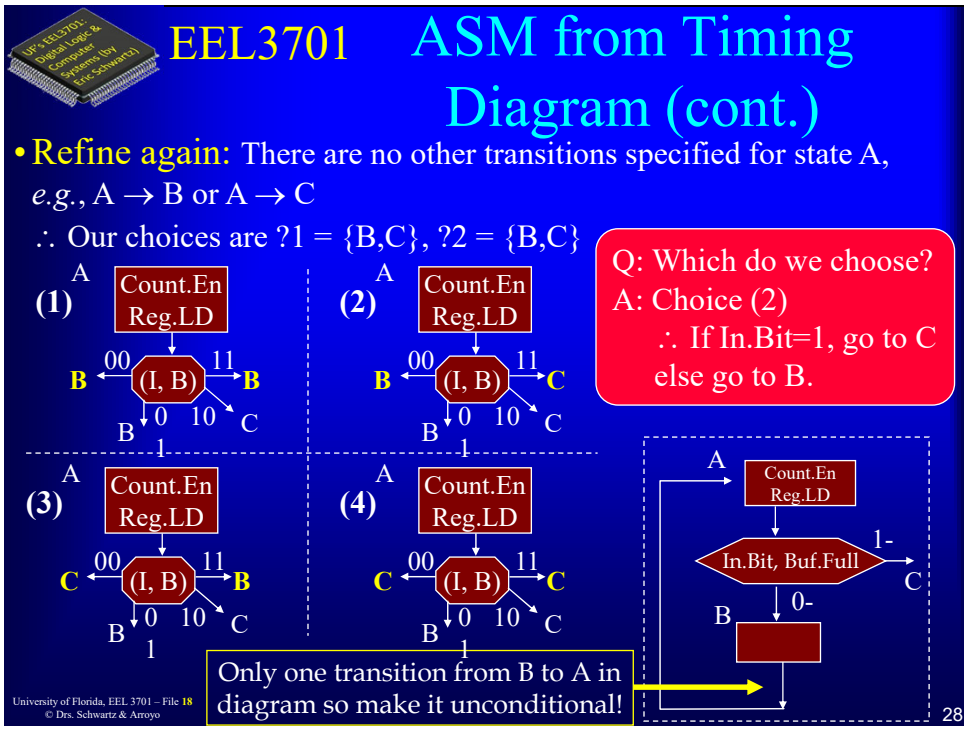
25


EEL3701 ASM To Timing Diagram
See Lam Fig 7.2(b)
T1 T2 T3 T4 T5 T6 T7
CLK
State
A B A C C D A
IN.BIT
BUF.FULL
COUNT.EN
REG.LD
OUT.FLAG
A 00
Count.En
Reg.LD
In.Bit
B 01
0
1
C 10
Reg.LD
0
Buf.Full
1
Count.En
D 11
Out.Flag
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
26

26


EEL3701
ASM from Timing Diagram
Skill 2: Given the timing diagram of Lam Fig 7.2(b) (on previous page) Produce an ASM
(a) Start at A: Count.En and Reg.LD are T, Inputs In.Bit= 0 & Buf.Full = 1, go to B.
(b) Now in B. No Outputs. Inputs In.Bit=0 & Buf.Full = 0, go to A.
(c) Back to A: Count.En and Reg.LD are T. Inputs In.Bit= 1 & Buf.Full = 0, go to C.
A
Count.En
Reg.LD
I,B
10
C
01
B
?
00
• Refine A, B
In.Bit,Buf.Full
11
?1
?2
?3
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
27

27

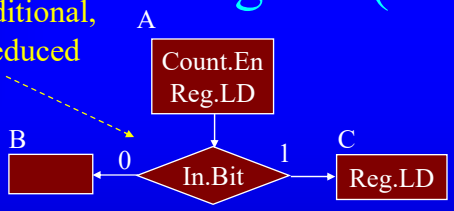

EEL3701
ASM from Timing Diagram (cont.)
• Refine again: There are no other transitions specified for state A,
e.g., A → B or A → C
∴ Our choices are ?1 = {B,C}, ?2 = {B,C}
Q: Which do we choose?
A: Choice (2)
∴ If In.Bit=1, go to C
else go to B.
(1)
(2)
(3)
(4)
Count.En
Reg.LD
(I, B)
00
11
10
01
Only one transition from B to A in diagram so make it unconditional!
In.Bit, Buf.Full
1-
0-
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
28

28

EEL3701

ASM from Timing Diagram (cont.)

- Since 0- or 1- in conditional, conditional can be reduced from 2 inputs to 1.



(d) Now in C. Reg.LD = T. Inputs In.Bit=1 & Buf.Full = 0, stay at C.

(e) Still in C. Count.En & Reg.LD are T. Inputs In.Bit= 1, Buf.Full = 1, go to D.

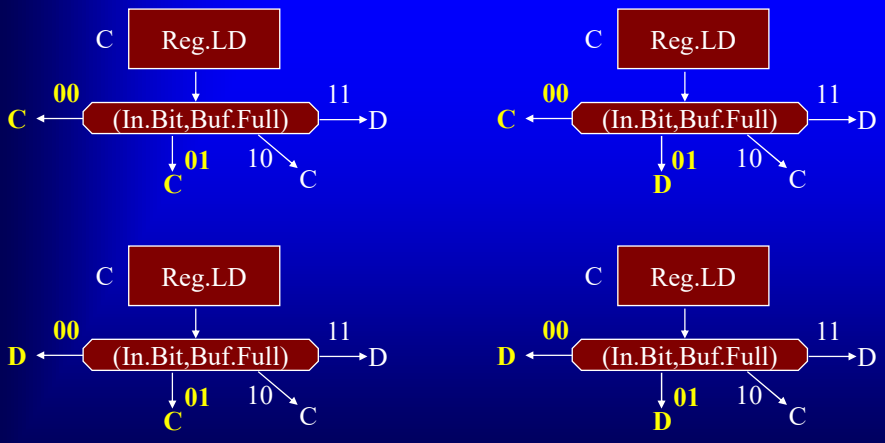
[NOTE: From (d) & (e), Count.En = T only when Buf.Full = 1. Therefore Count.En is a conditional output!]

- Again, note that input combination {0,0} and {0,1} are not specified. What was specified was that if Buf.Full = T, go to D, else go to C. By the same analysis as for state A, we choose the simplest possibility.

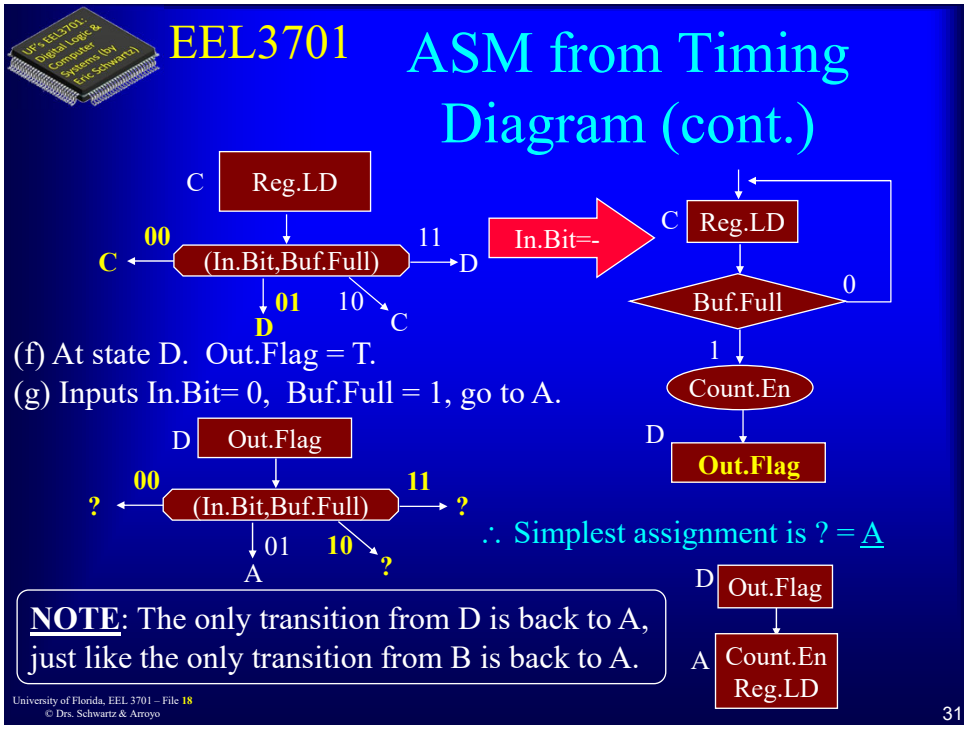
29

EEL3701

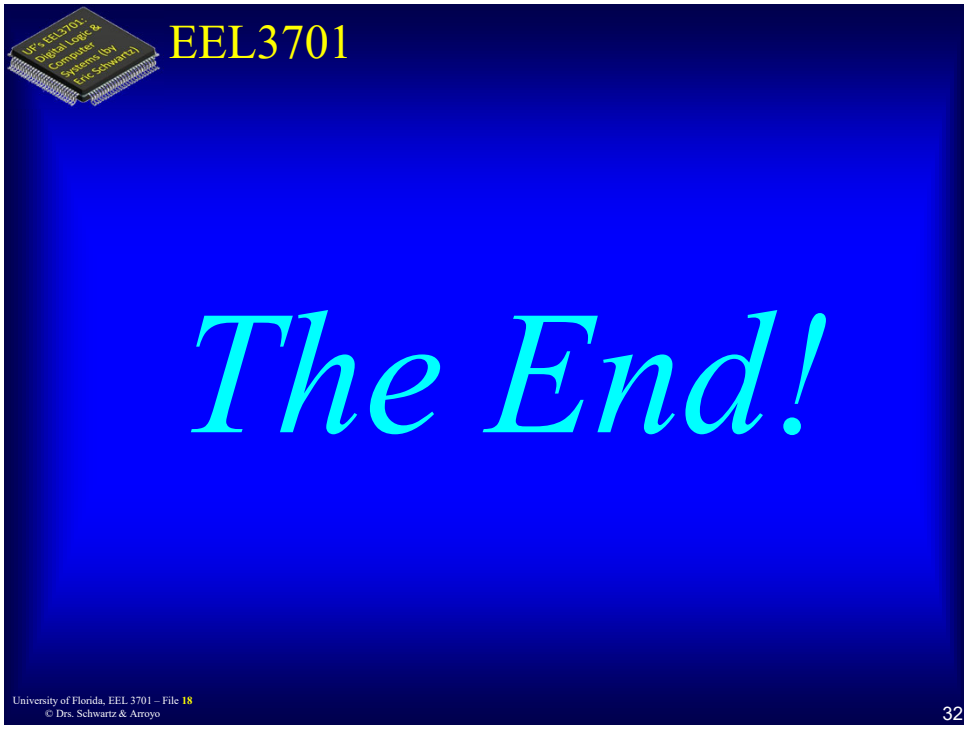
ASM from Timing Diagram (cont.)



30


EEL3701
ASM from Timing Diagram (cont.)
C
Reg.LD
00
11
C
(In.Bit,Buf.Full)
D
01
10
D
C
In.Bit=-
C
Reg.LD
Buf.Full
0
1
Count.En
D
Out.Flag
(f) At state D. Out.Flag = T.
(g) Inputs In.Bit= 0, Buf.Full = 1, go to A.
D
Out.Flag
00
11
?
(In.Bit,Buf.Full)
?
01
10
A
?
∴ Simplest assignment is ? = A
D
Out.Flag
A
Count.En
Reg.LD
NOTE: The only transition from D is back to A, just like the only transition from B is back to A.
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
31

31


EEL3701
The End!
University of Florida, EEL 3701 – File 18
© Drs. Schwartz & Arroyo
32

32