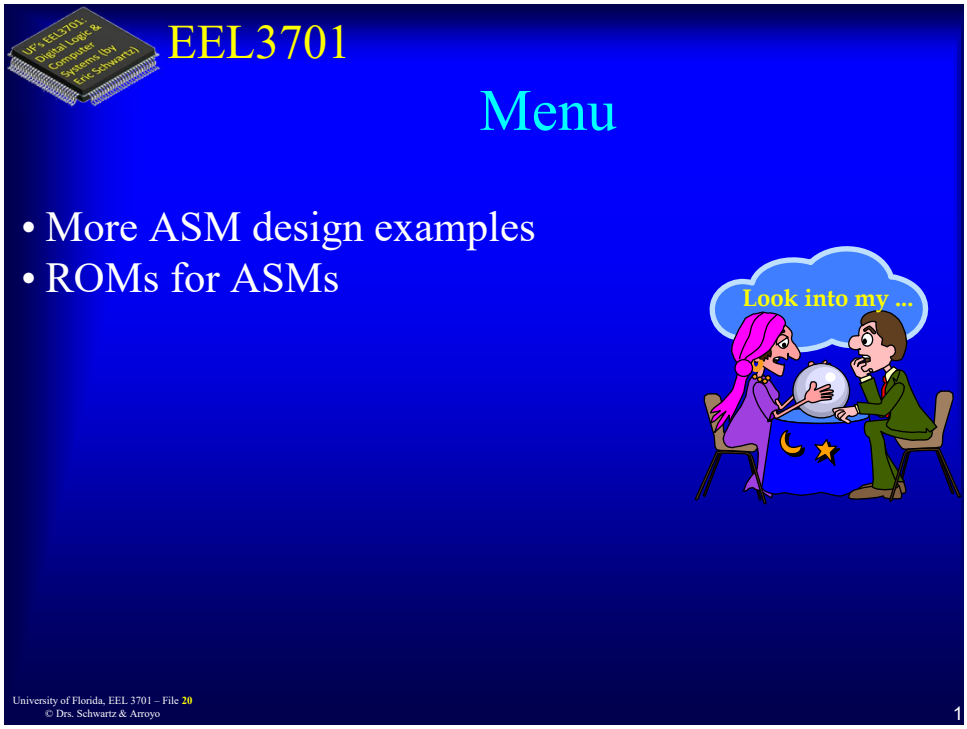
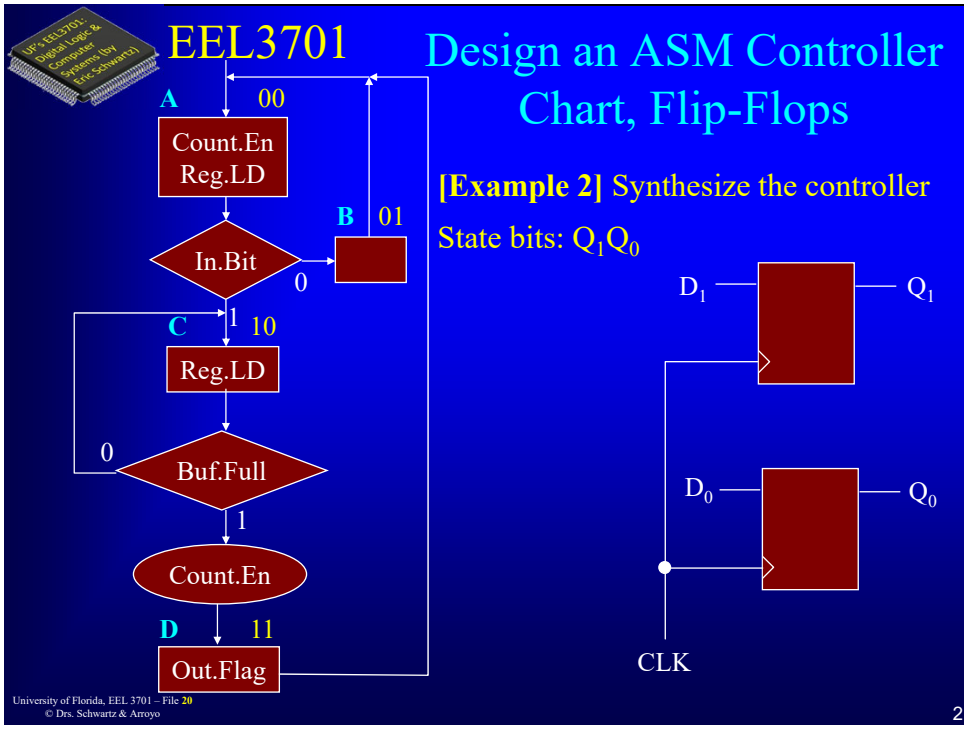
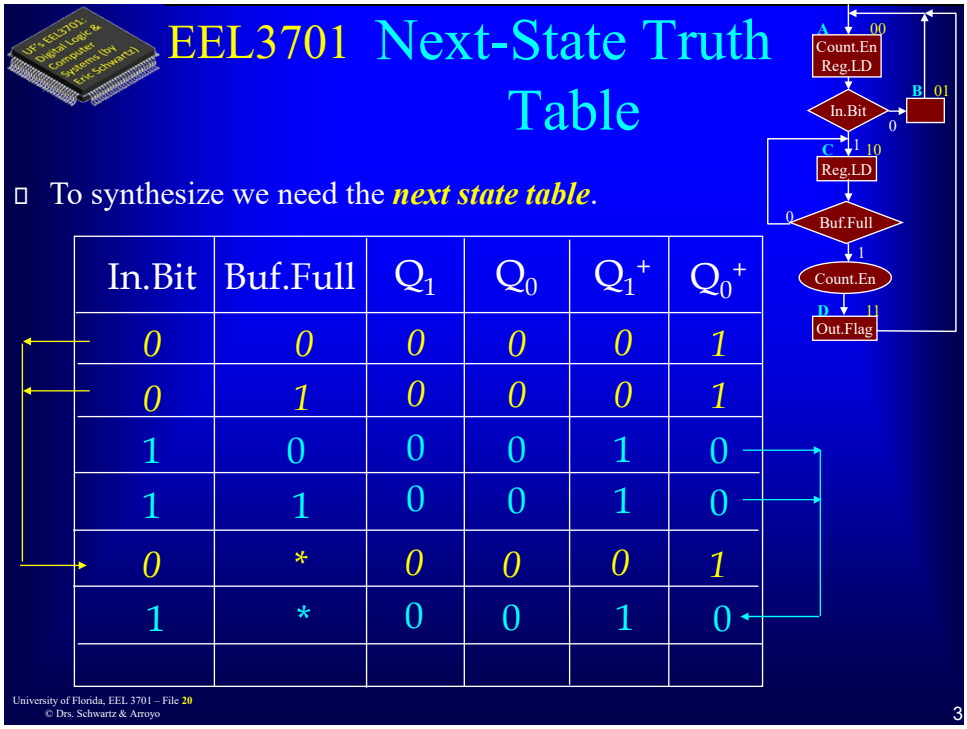

EEL3701
Menu
• More ASM design examples
• ROMs for ASMs
Look into my ...
University of Florida, EEL 3701 – File 20
© Drs. Schwartz & Arroyo
1

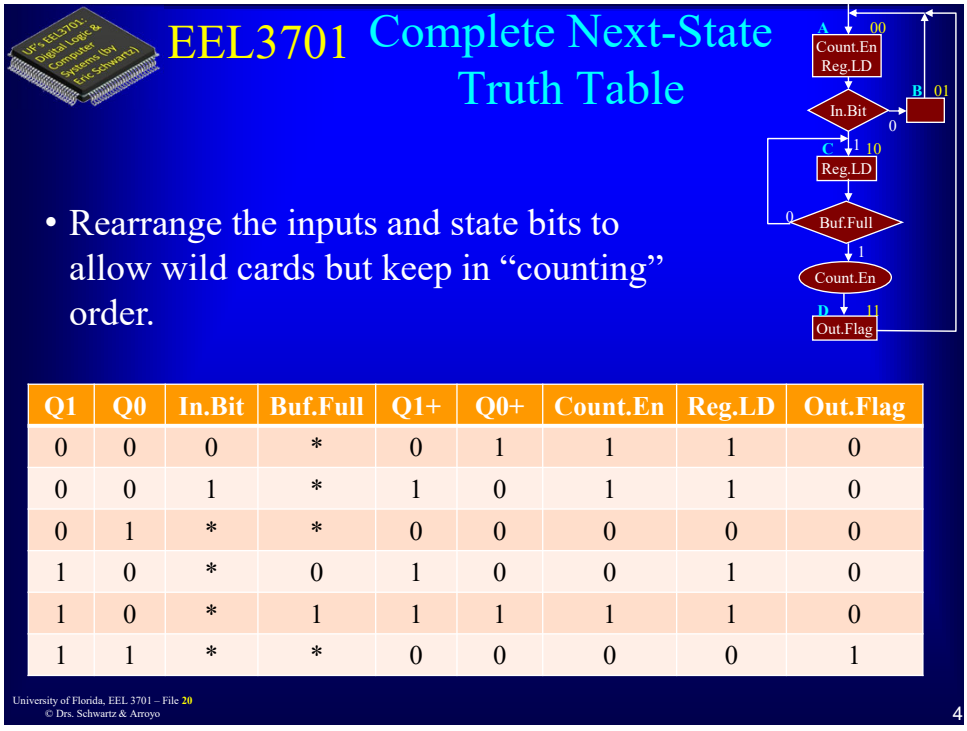
1


EEL3701
Design an ASM Controller Chart, Flip-Flops
[Example 2] Synthesize the controller
State bits: Q_1Q_0
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
D_1
Q_1
D_0
Q_0
CLK
University of Florida, EEL 3701 – File 20
© Drs. Schwartz & Arroyo
2

2

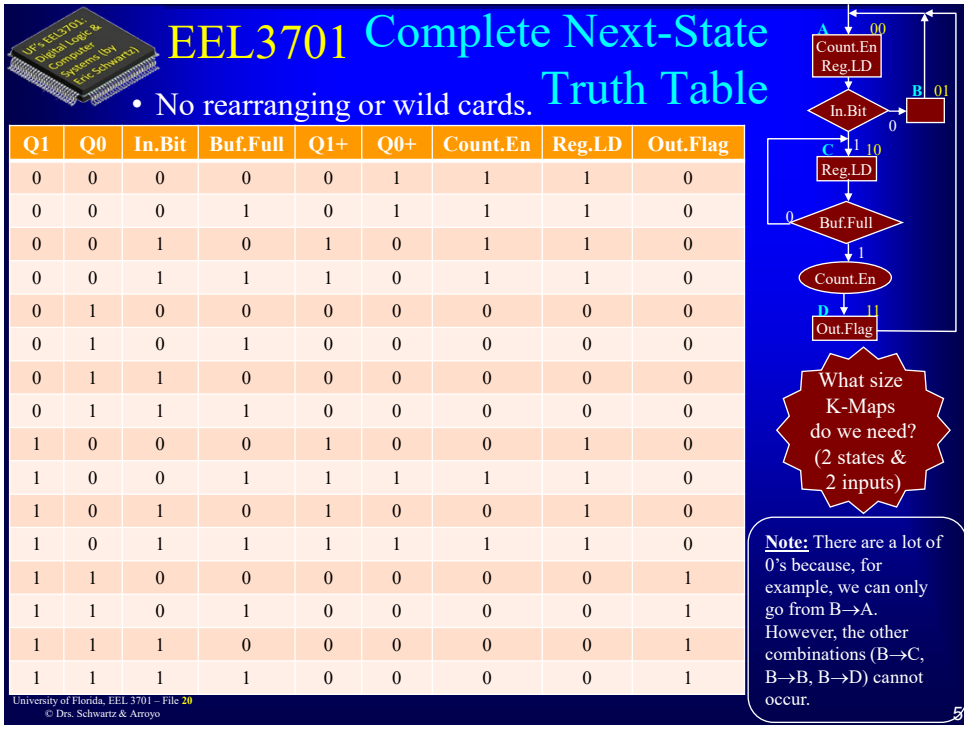


3



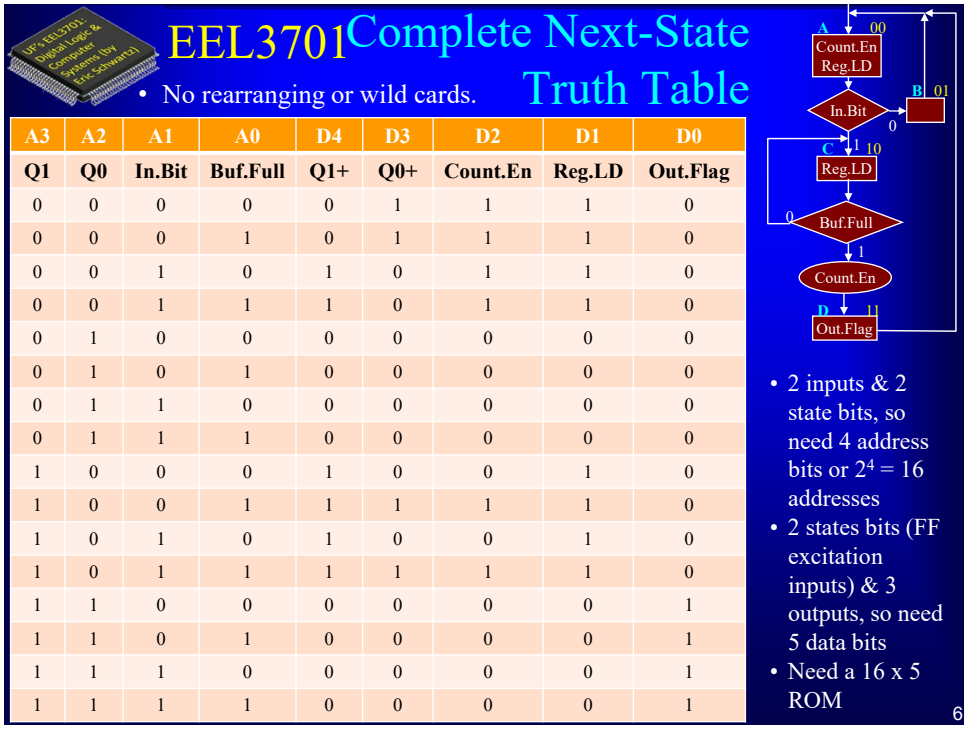
Q1	Q0	In.Bit	Buf.Full	Q1+	Q0+	Count.En	Reg.LD	Out.Flag
0	0	0	*	0	1	1	1	0
0	0	1	*	1	0	1	1	0
0	1	*	*	0	0	0	0	0
1	0	*	0	1	0	0	1	0
1	0	*	1	1	1	1	1	0
1	1	*	*	0	0	0	0	1

4



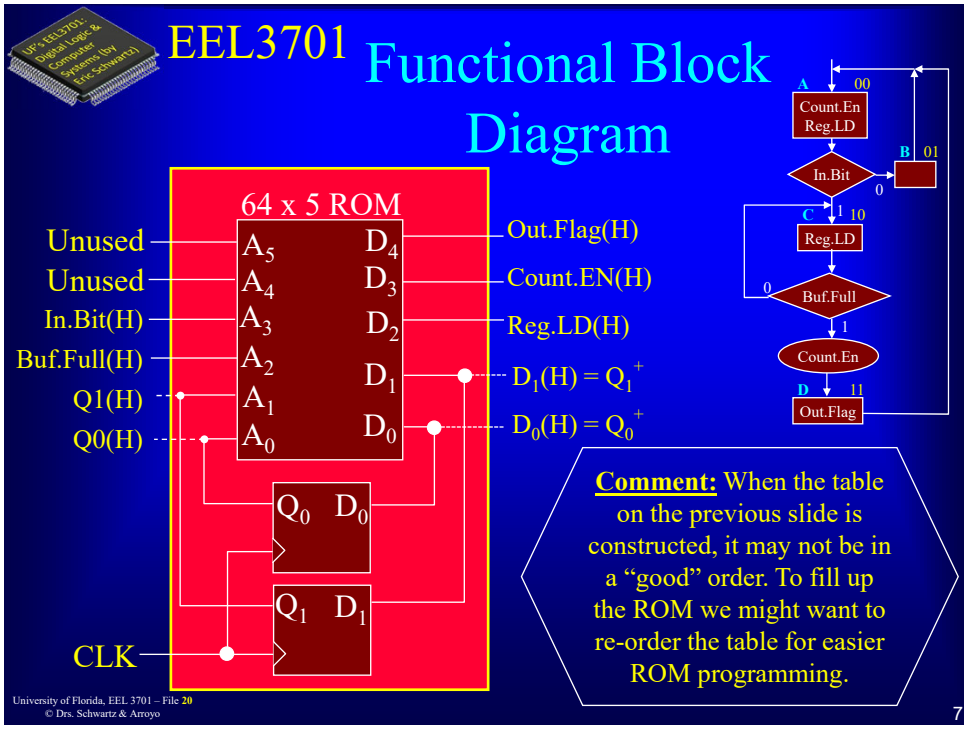
Q1	Q0	In.Bit	Buf.Full	Q1+	Q0+	Count.En	Reg.LD	Out.Flag
0	0	0	0	0	1	1	1	0
0	0	0	1	0	1	1	1	0
0	0	1	0	1	0	1	1	0
0	0	1	1	1	0	1	1	0
0	1	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0
1	0	0	0	1	0	0	1	0
1	0	0	1	1	1	1	1	0
1	0	1	0	1	0	0	1	0
1	0	1	1	1	1	1	1	0
1	1	0	0	0	0	0	0	1
1	1	0	1	0	0	0	0	1
1	1	1	0	0	0	0	0	1
1	1	1	1	0	0	0	0	1

5

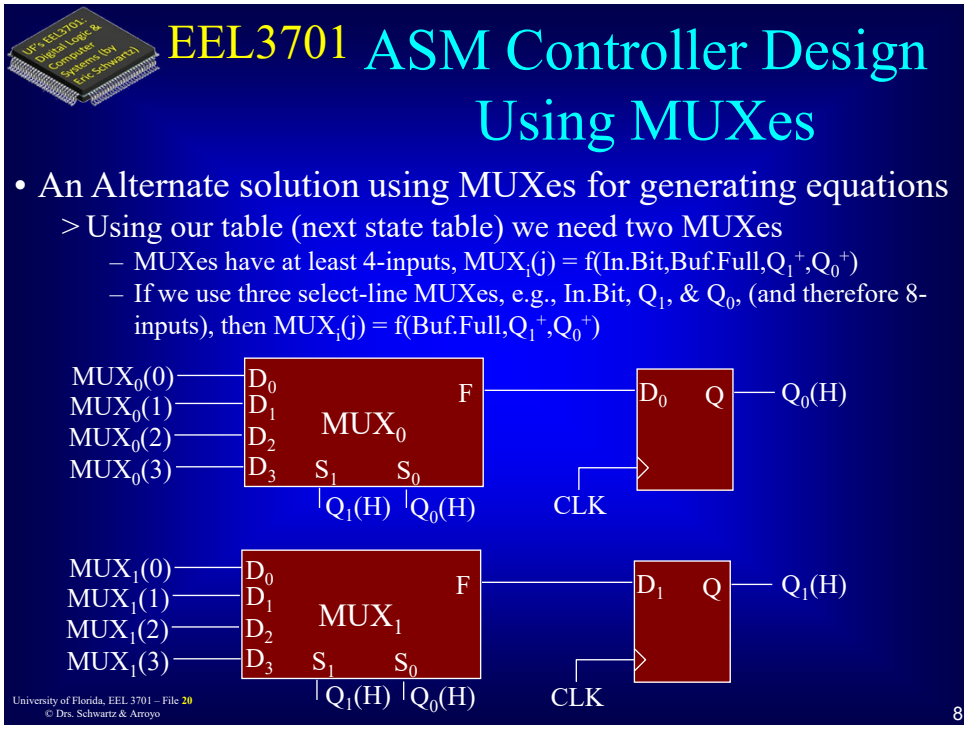


A3	A2	A1	A0	D4	D3	D2	D1	D0
Q1	Q0	In.Bit	Buf.Full	Q1+	Q0+	Count.En	Reg.LD	Out.Flag
0	0	0	0	0	1	1	1	0
0	0	0	1	0	1	1	1	0
0	0	1	0	1	0	1	1	0
0	0	1	1	1	0	1	1	0
0	1	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0
1	0	0	0	1	0	0	1	0
1	0	0	1	1	1	1	1	0
1	0	1	0	1	0	0	1	0
1	0	1	1	1	1	1	1	0
1	1	0	0	0	0	0	0	1
1	1	0	1	0	0	0	0	1
1	1	1	0	0	0	0	0	1
1	1	1	1	0	0	0	0	1

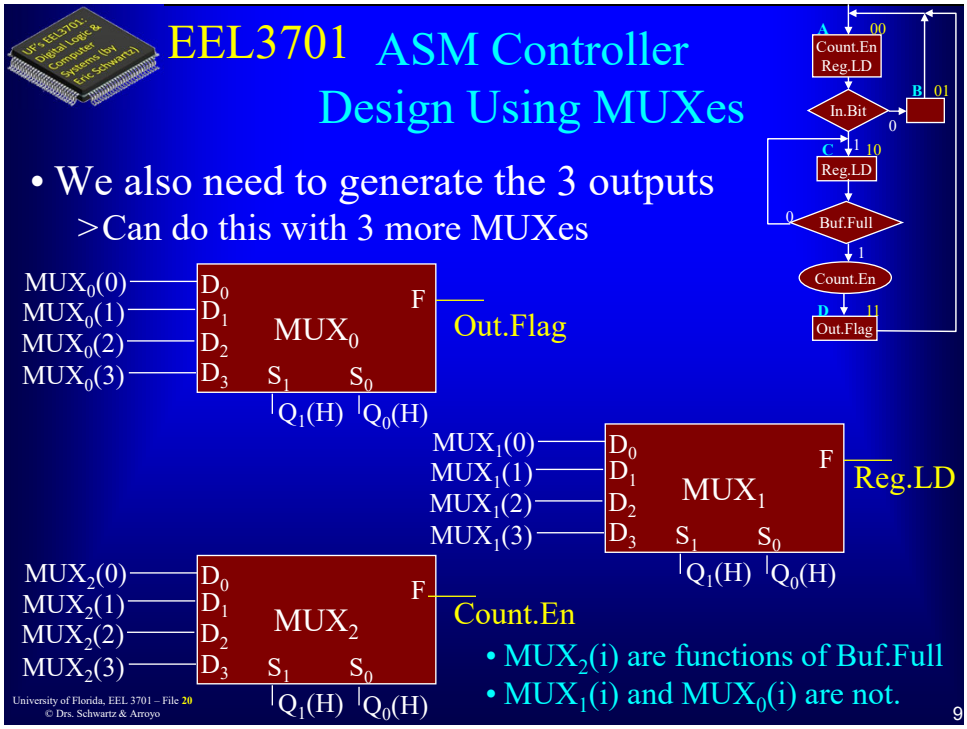
6


EEL3701
Functional Block Diagram
64 x 5 ROM
Unused
Unused
In.Bit(H)
Buf.Full(H)
Q1(H)
Q0(H)
CLK
A5
A4
A3
A2
A1
A0
D4
D3
D2
D1
D0
Out.Flag(H)
Count.EN(H)
Reg.LD(H)
D1(H) = Q1+
D0(H) = Q0+
Q0 D0
Q1 D1
A 00
Count.En
Reg.LD
B 01
In.Bit
0
C 1 10
Reg.LD
0
Buf.Full
1
Count.En
D 11
Out.Flag
Comment: When the table on the previous slide is constructed, it may not be in a "good" order. To fill up the ROM we might want to re-order the table for easier ROM programming.
University of Florida, EEL 3701 – File 20
© Drs. Schwartz & Arroyo
7

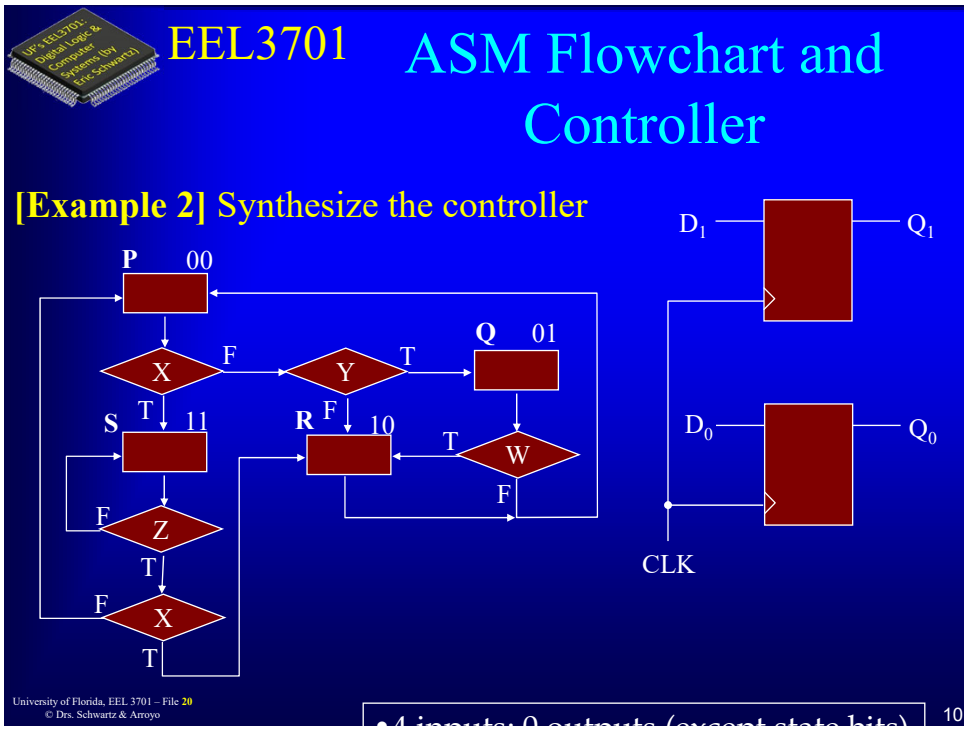
7


EEL3701
ASM Controller Design Using MUXes
• An Alternate solution using MUXes for generating equations
> Using our table (next state table) we need two MUXes
– MUXes have at least 4-inputs, MUXi(j) = f(In.Bit,Buf.Full,Q1+,Q0+)
– If we use three select-line MUXes, e.g., In.Bit, Q1, & Q0, (and therefore 8-inputs), then MUXi(j) = f(Buf.Full,Q1+,Q0+)
MUX0(0)
MUX0(1)
MUX0(2)
MUX0(3)
MUX0
F
S1 S0
Q1(H) Q0(H)
D0 Q
Q0(H)
CLK
MUX1(0)
MUX1(1)
MUX1(2)
MUX1(3)
MUX1
F
S1 S0
Q1(H) Q0(H)
D1 Q
Q1(H)
CLK
University of Florida, EEL 3701 – File 20
© Drs. Schwartz & Arroyo
8

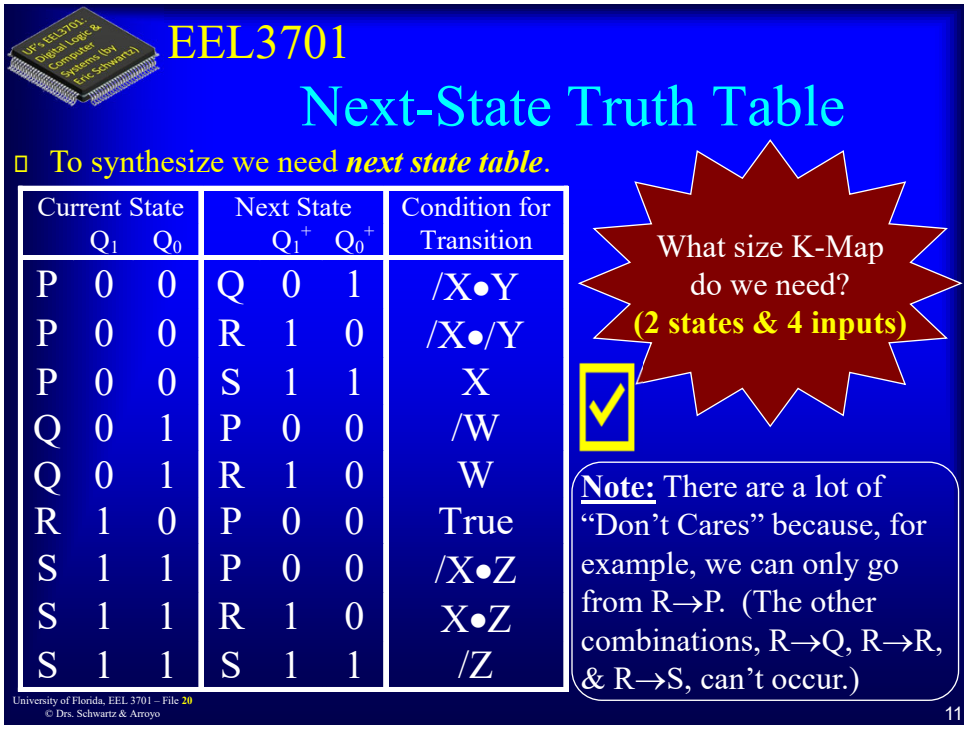
8


EEL3701
ASM Controller Design Using MUXes
• We also need to generate the 3 outputs
>Can do this with 3 more MUXes
MUX0(0)
MUX0(1)
MUX0(2)
MUX0(3)
D0
D1
D2
D3
MUX0
F
Out.Flag
S1
S0
Q1(H)
Q0(H)
MUX1(0)
MUX1(1)
MUX1(2)
MUX1(3)
MUX1
Reg.LD
MUX2(0)
MUX2(1)
MUX2(2)
MUX2(3)
MUX2
Count.En
• MUX2(i) are functions of Buf.Full
• MUX1(i) and MUX0(i) are not.
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 20
© Drs. Schwartz & Arroyo
9

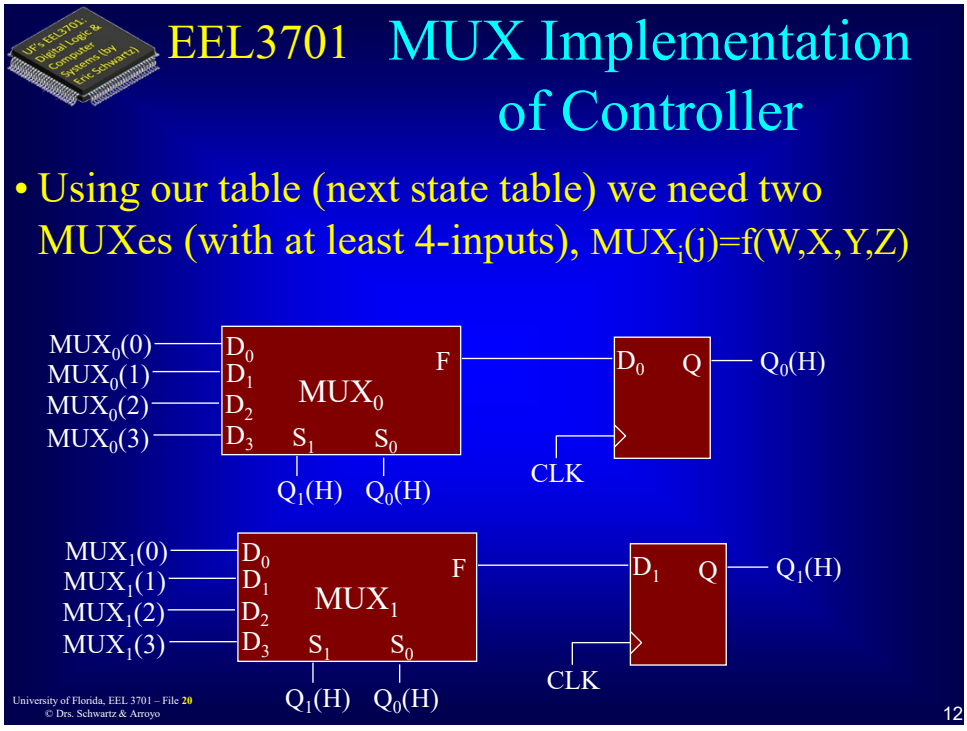
9


EEL3701
ASM Flowchart and Controller
[Example 2] Synthesize the controller
P 00
X
F
Y
T
Q 01
S
T
11
R
F
10
T
W
F
F
Z
T
F
X
T
D1
Q1
D0
Q0
CLK
University of Florida, EEL 3701 – File 20
© Drs. Schwartz & Arroyo
10

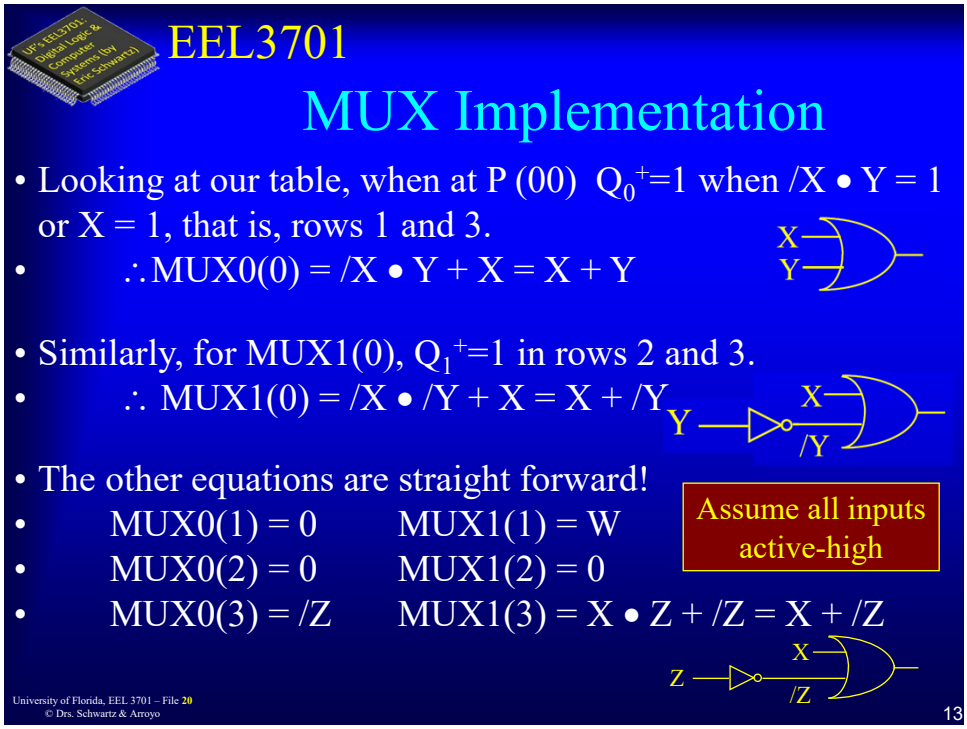
10


EEL3701
Next-State Truth Table
To synthesize we need *next state table*.
Current State Q1 Q0 | Next State Q1+ Q0+ | Condition for Transition
P 0 0 | Q 0 1 | /X•Y
P 0 0 | R 1 0 | /X•/Y
P 0 0 | S 1 1 | X
Q 0 1 | P 0 0 | /W
Q 0 1 | R 1 0 | W
R 1 0 | P 0 0 | True
S 1 1 | P 0 0 | /X•Z
S 1 1 | R 1 0 | X•Z
S 1 1 | S 1 1 | /Z
What size K-Map do we need?
(2 states & 4 inputs)
Note: There are a lot of "Don't Cares" because, for example, we can only go from R→P. (The other combinations, R→Q, R→R, & R→S, can't occur.)
University of Florida, EEL 3701 – File 20
© Drs. Schwartz & Arroyo
11

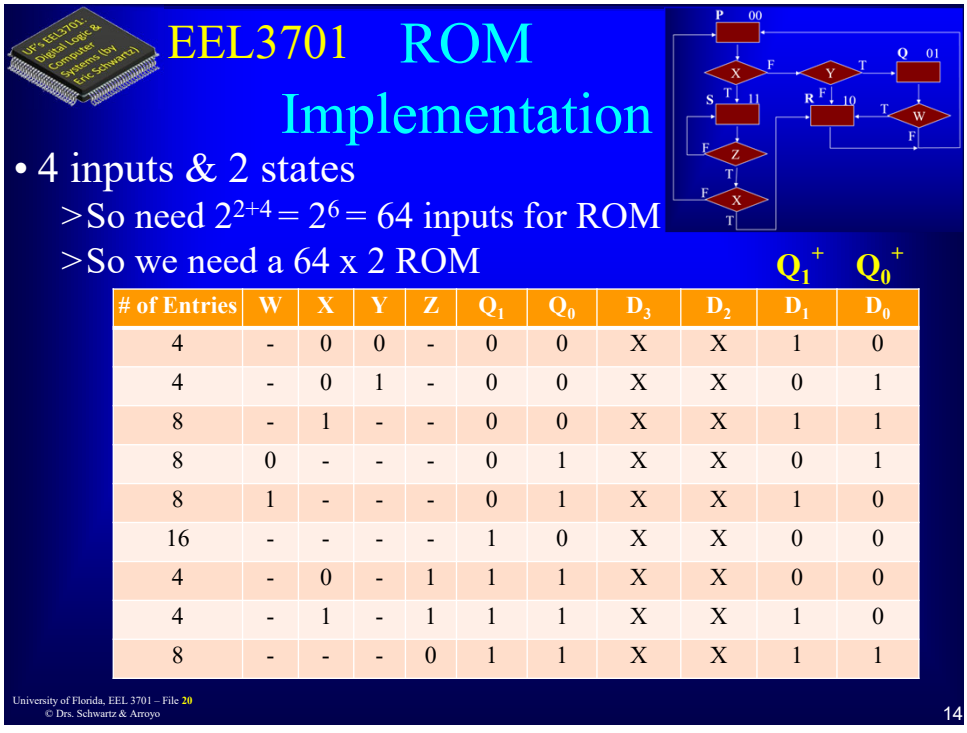
11


EEL3701 MUX Implementation of Controller
• Using our table (next state table) we need two MUXes (with at least 4-inputs), MUXi(j)=f(W,X,Y,Z)
MUX0(0)
MUX0(1)
MUX0(2)
MUX0(3)
D0 D1 D2 D3
MUX0
F
S1 S0
Q1(H) Q0(H)
D0 Q
Q0(H)
CLK
MUX1(0)
MUX1(1)
MUX1(2)
MUX1(3)
D0 D1 D2 D3
MUX1
F
S1 S0
Q1(H) Q0(H)
D1 Q
Q1(H)
CLK
University of Florida, EEL 3701 – File 20
© Drs. Schwartz & Arroyo
12

12

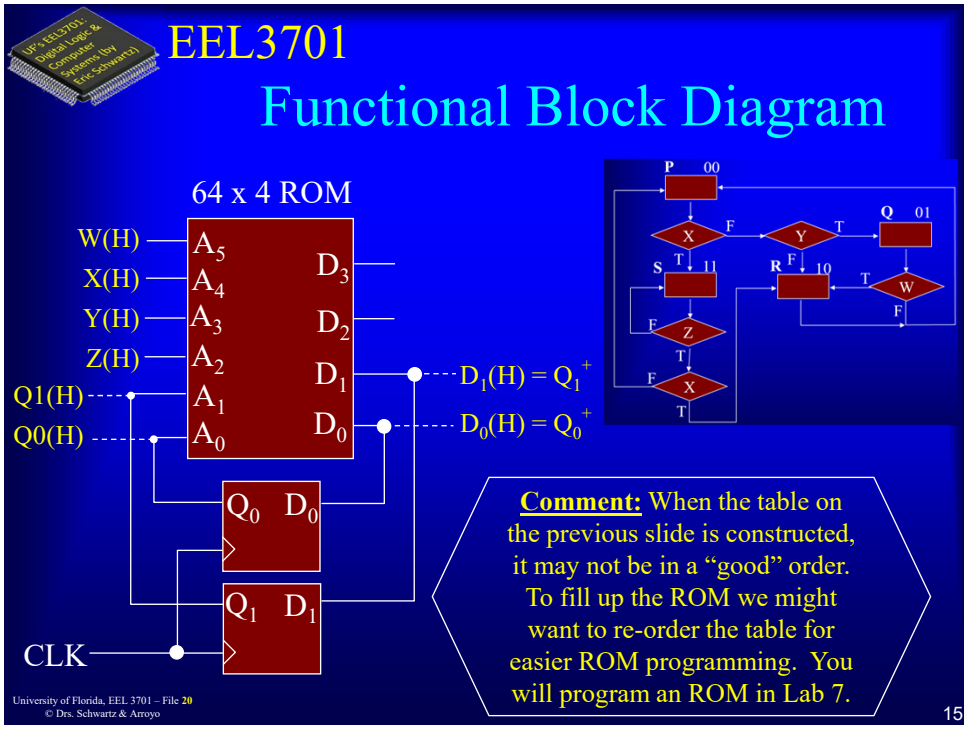


13

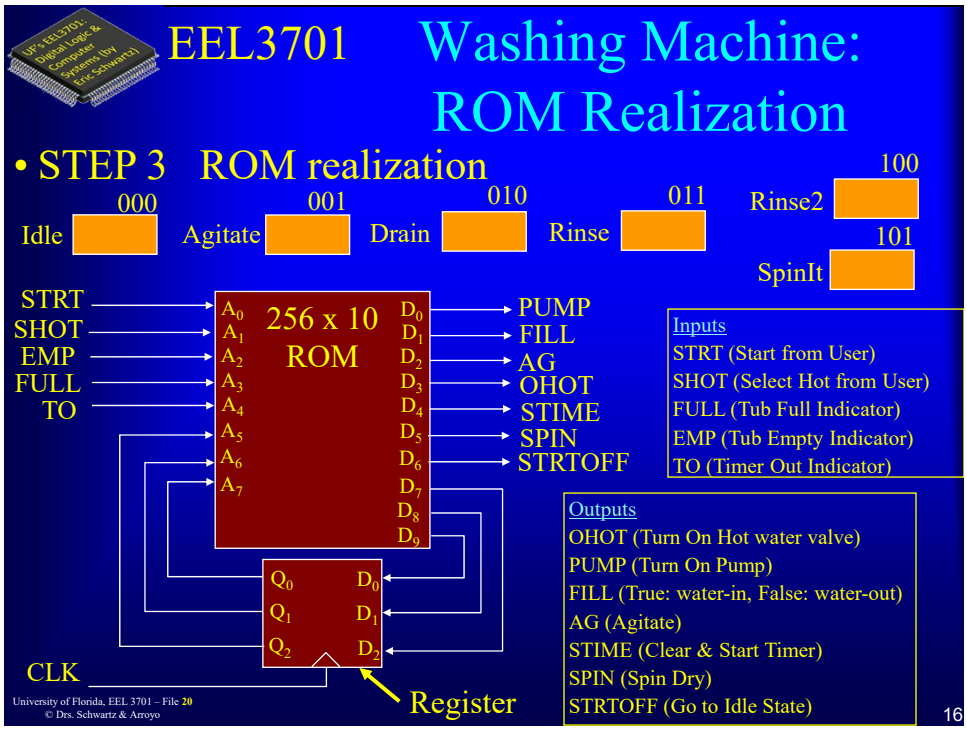


# of Entries	W	X	Y	Z	Q_1	Q_0	D_3	D_2	D_1	D_0
4	-	0	0	-	0	0	X	X	1	0
4	-	0	1	-	0	0	X	X	0	1
8	-	1	-	-	0	0	X	X	1	1
8	0	-	-	-	0	1	X	X	0	1
8	1	-	-	-	0	1	X	X	1	0
16	-	-	-	-	1	0	X	X	0	0
4	-	0	-	1	1	1	X	X	0	0
4	-	1	-	1	1	1	X	X	1	0
8	-	-	-	0	1	1	X	X	1	1

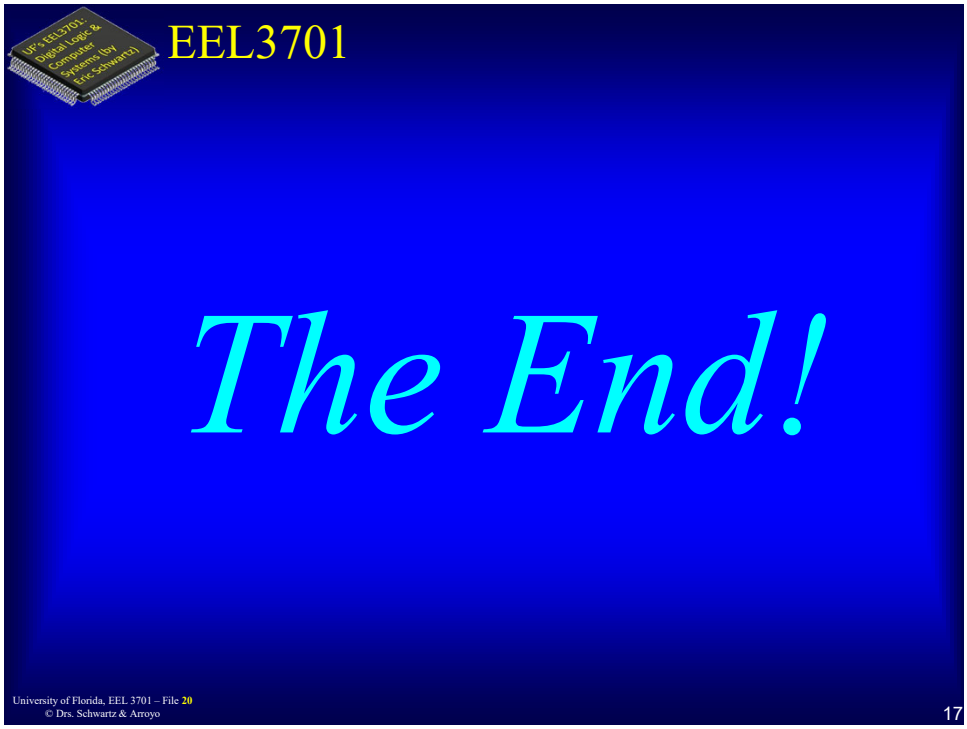
14


EEL3701
Functional Block Diagram
64 x 4 ROM
W(H)
X(H)
Y(H)
Z(H)
Q1(H)
Q0(H)
A5
A4
A3
A2
A1
A0
D3
D2
D1
D0
D1(H) = Q1+
D0(H) = Q0+
Q0 D0
Q1 D1
CLK
P 00
Q 01
X
Y
S 11
R 10
W
Z
Comment: When the table on the previous slide is constructed, it may not be in a "good" order. To fill up the ROM we might want to re-order the table for easier ROM programming. You will program an ROM in Lab 7.
University of Florida, EEL 3701 – File 20
© Drs. Schwartz & Arroyo
15

15


EEL3701
Washing Machine:
ROM Realization
• STEP 3 ROM realization
Idle 000
Agitate 001
Drain 010
Rinse 011
Rinse2 100
SpinIt 101
STRT
SHOT
EMP
FULL
TO
256 x 10 ROM
PUMP
FILL
AG
OHOT
STIME
SPIN
STRTOFF
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)
CLK
Register
University of Florida, EEL 3701 – File 20
© Drs. Schwartz & Arroyo
16

16


EEL3701
The End!
University of Florida, EEL 3701 – File 20
© Drs. Schwartz & Arroyo
17

17