



CSCE 611

Digital Systems Design I

2005/3/27

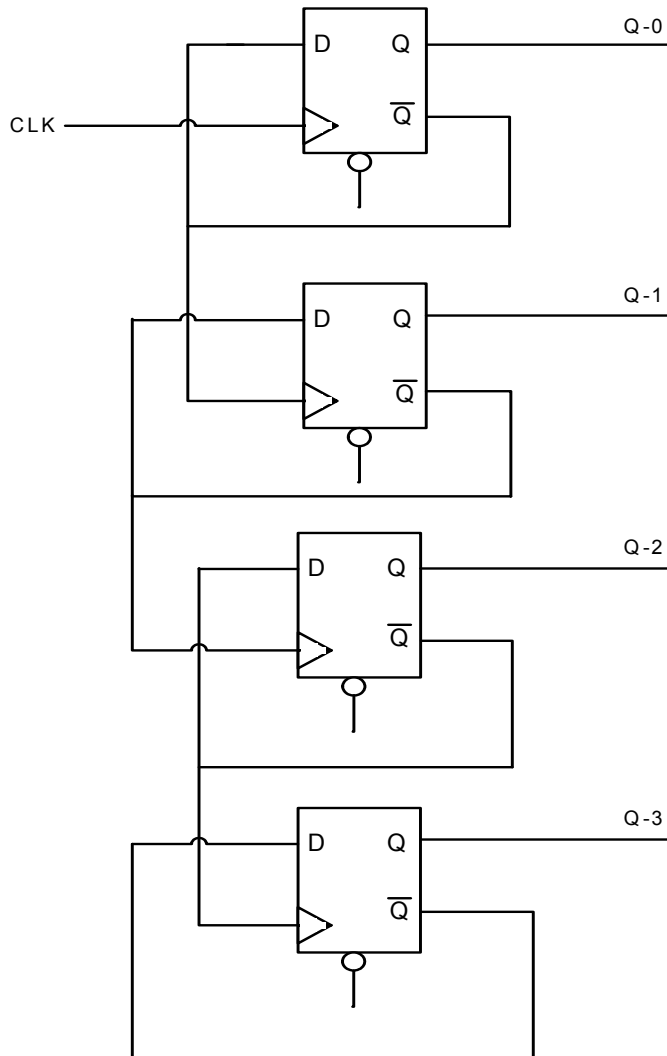
Week 8 (Design Lab)
ASM Design/Analysis using Logic Synthesis

© 2002 Dr. James P. Davis

Exercise Outline

- Introduction.
 - ✓ The Ripple and Ring Counters – analysis and design, and logic synthesis to evaluate architectures.
 - ✓ This is a lead-in to the Lab session for Friday's class.
- Analysis & Design methods.
 - ✓ Gate-level sequential machine analysis.
 - ✓ Gate-level to Register-level abstraction.
 - ✓ Register-level design.
 - ✓ Logic Synthesis and comparison of different design architectures.
- Lab
 - ✓ Take each of the three design architectures discussed in class, create models and evaluate the circuit architectures using logic synthesis and layout for the Xilinx Spartan-IIe®.

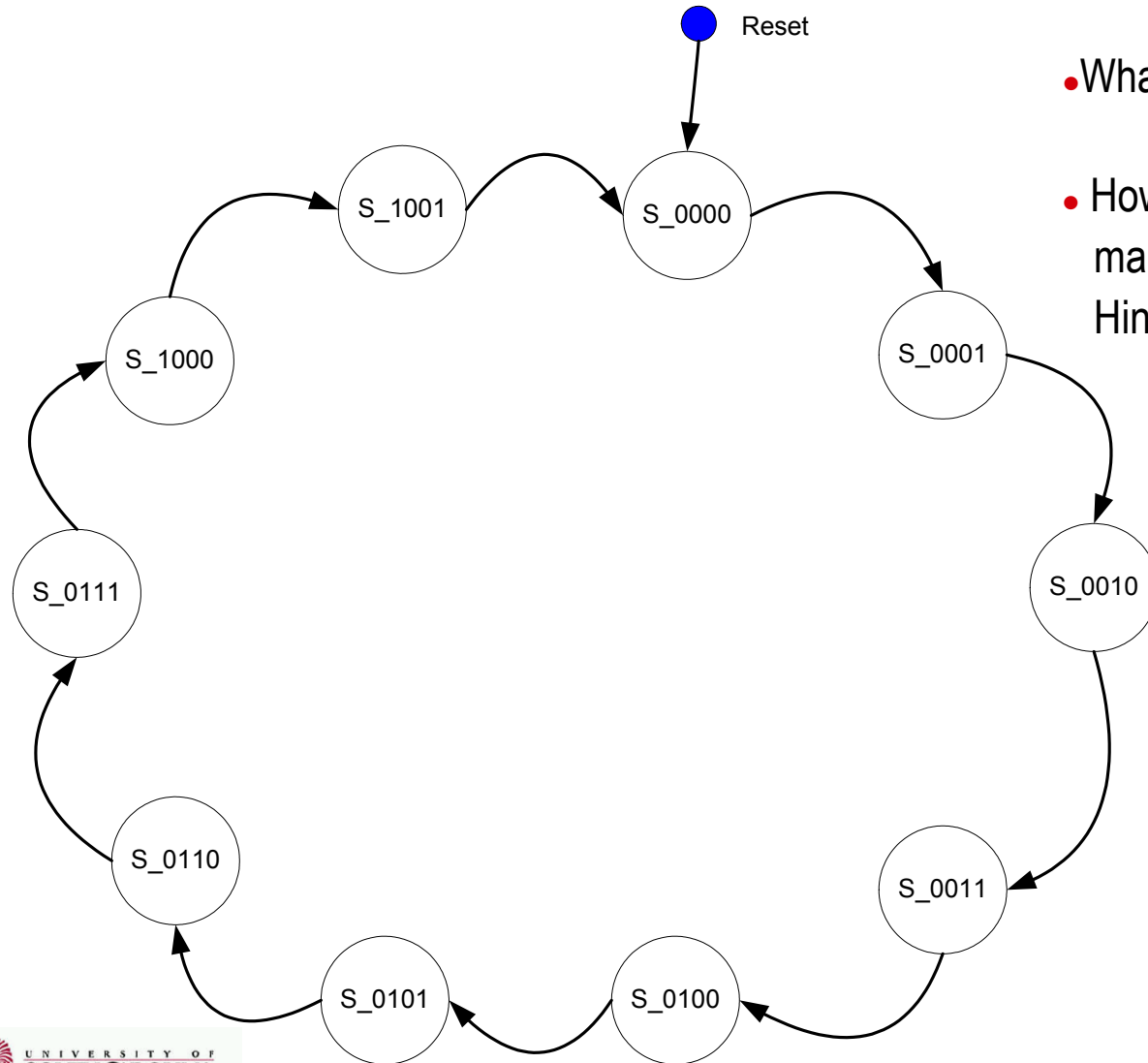
Lab Exercise – Modulo Counter Design



- Gate-level counter:

- ✓ Modulo counters can be implemented using D-Flip Flop “ripple chains”.
- ✓ A single-bit register FF encodes and outputs one of n binary digits, from 0 to digit $n-1$.
- ✓ Arrangement of the bit registers has inverted output \overline{Q}_k of lower counter FF k feeding input of register FF k and clock input of FF $k+1$, up to $n-1^{th}$ FF location $\{k = 0 \text{ to } n-1\}$.
- ✓ The inverted $n-1^{th}$ FF is fed back to its D input (inverting its input each time it is clocked).

Mod-10 Ripple Counter – Architecture #1

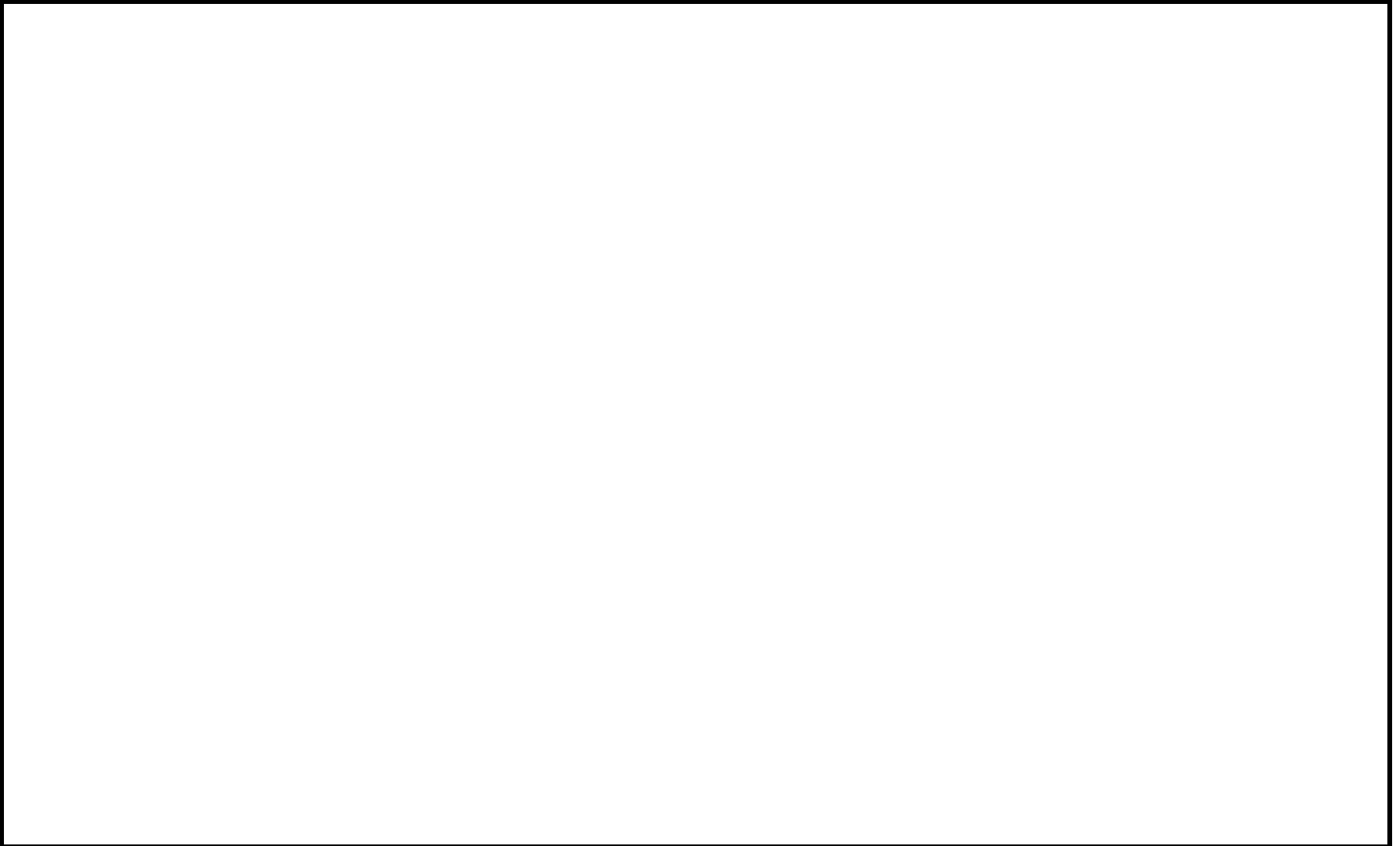


- What about the unused states?
- How would we change this to make it more generic (i.e., Mod-n)? Hint, see Architecture #2.

Laboratory ASM Worksheet

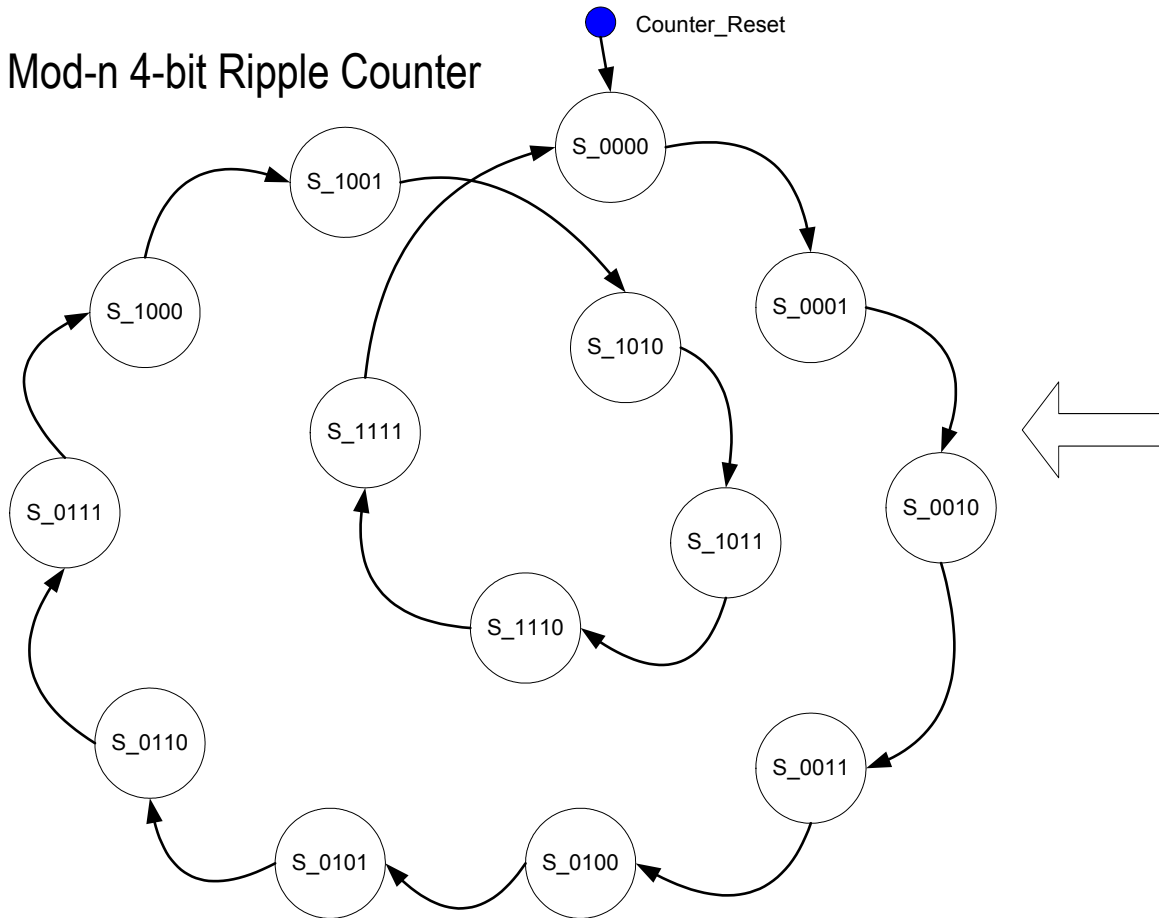
Thread_1

- Use this worksheet to hand-draw your initial ASM design.

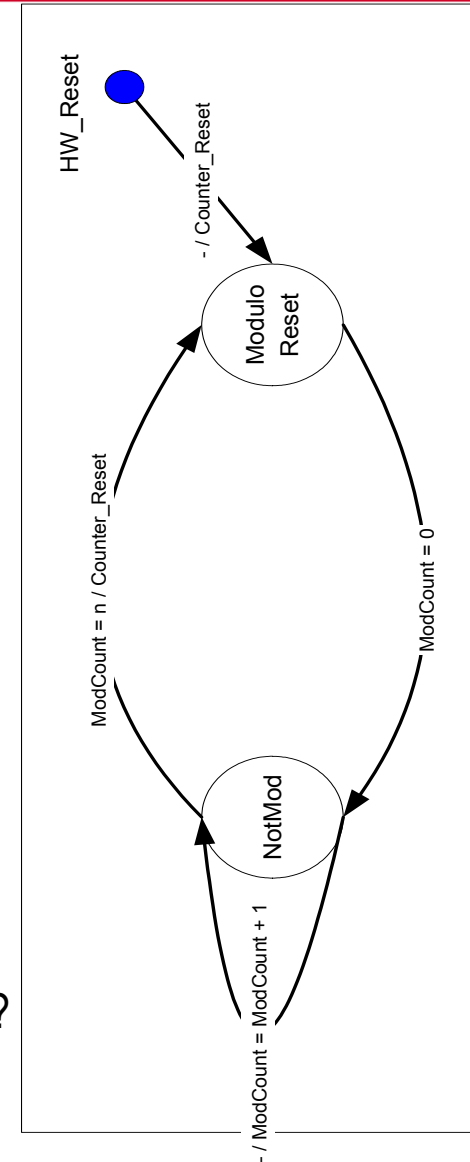


Progr. Mod-n Ripple Counter – Architecture #2

- Mod-n 4-bit Ripple Counter



- The second state machine controls the first one. How does it do it?
- How do the transition conditions in Thread 2 affect transitions in Thread 1?

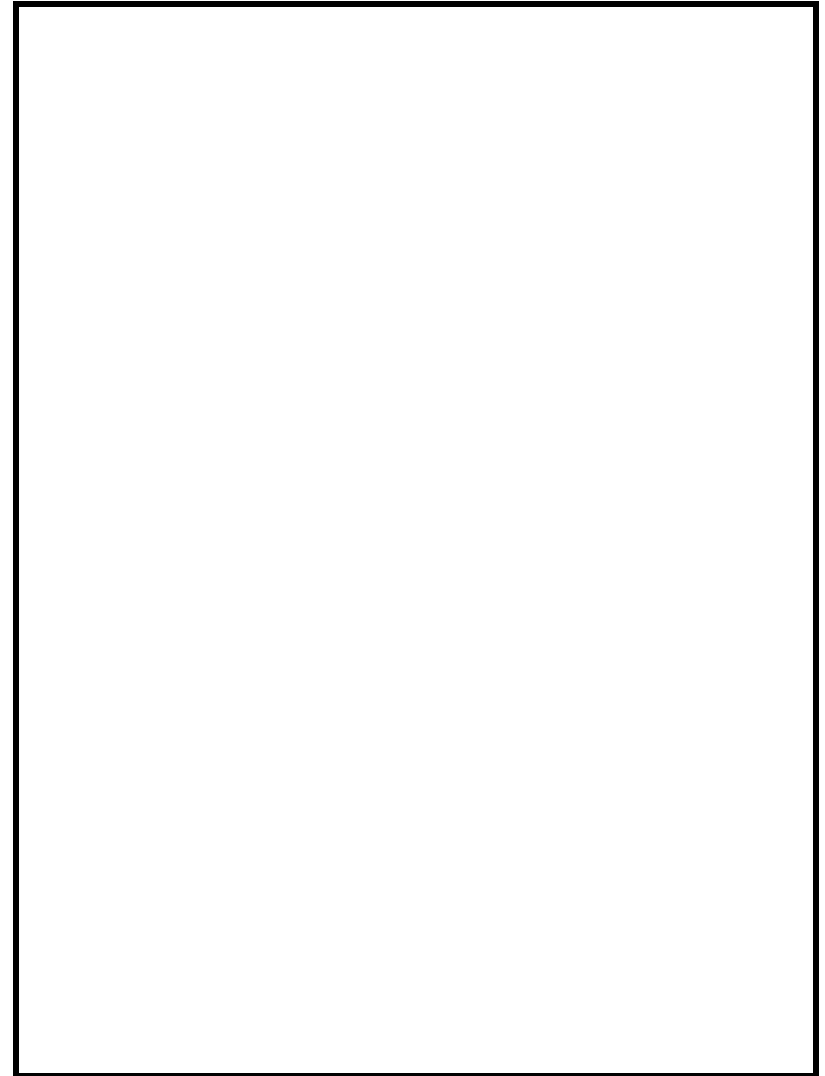


Laboratory ASM Worksheet

Thread_1



Thread_2

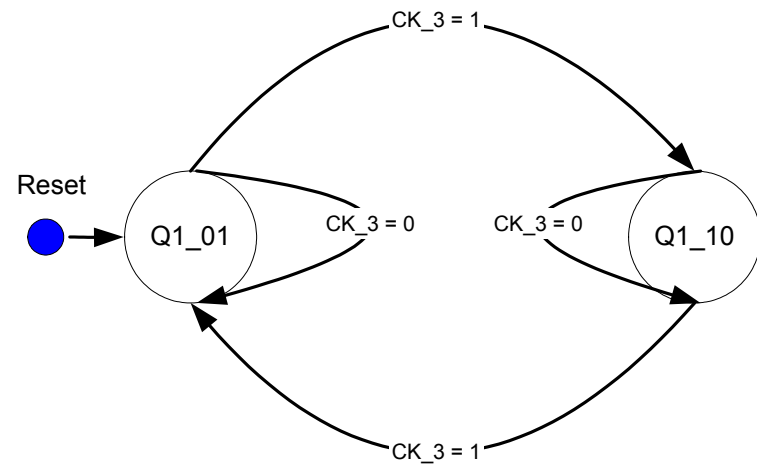
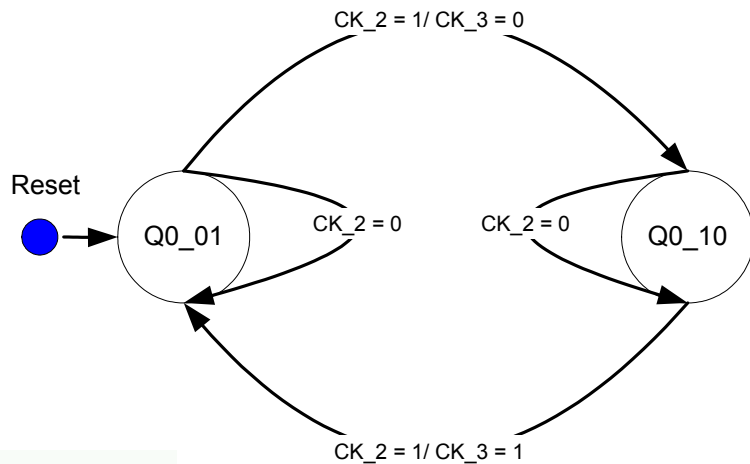
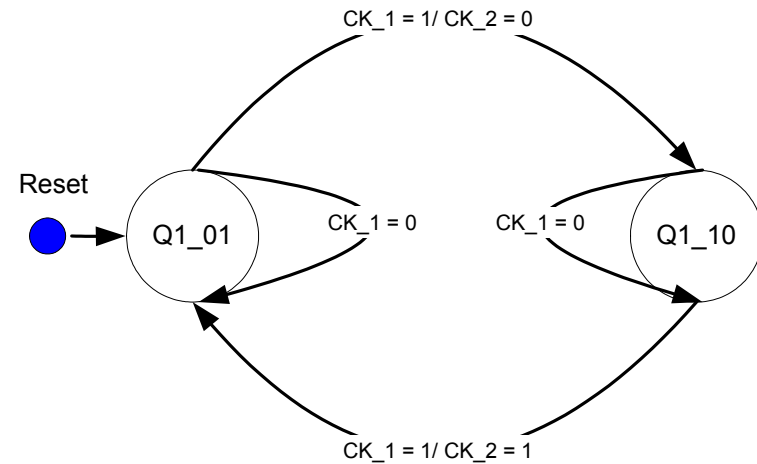
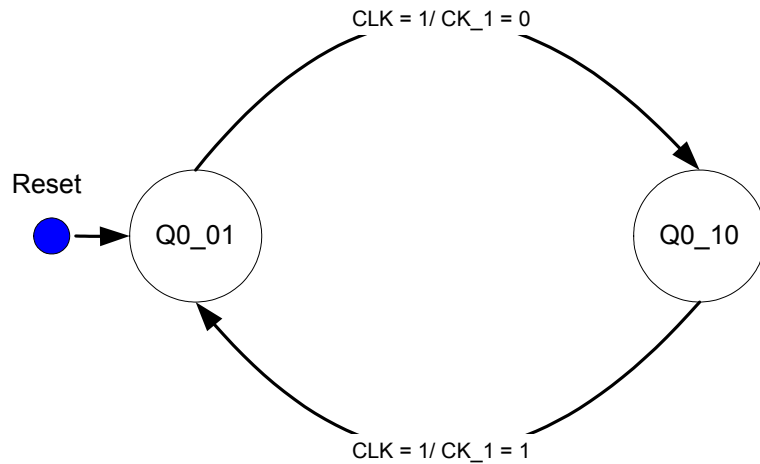


Use this worksheet to hand-draw your initial partitioned ASM design.

Show signal interconnection between the boxes.

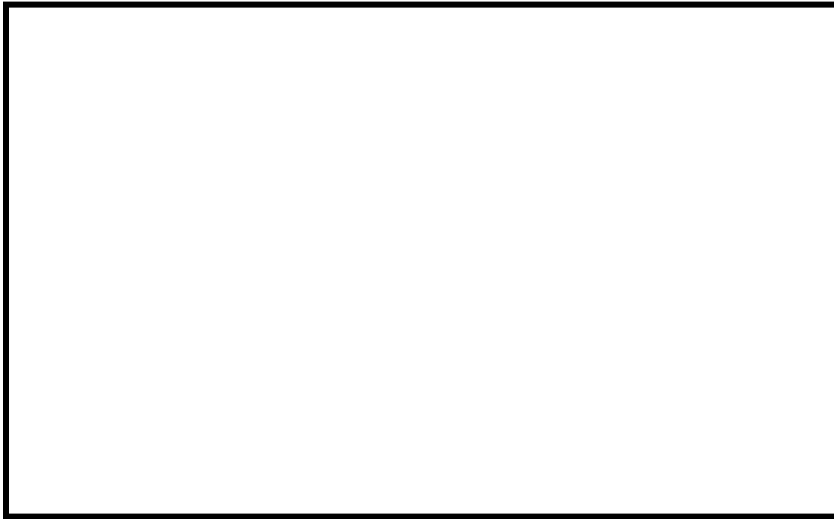
Concurrent Mod-n Ripple Counter – Architecture #3

- The Nth state machine controls the N+1 one. How does it do it?

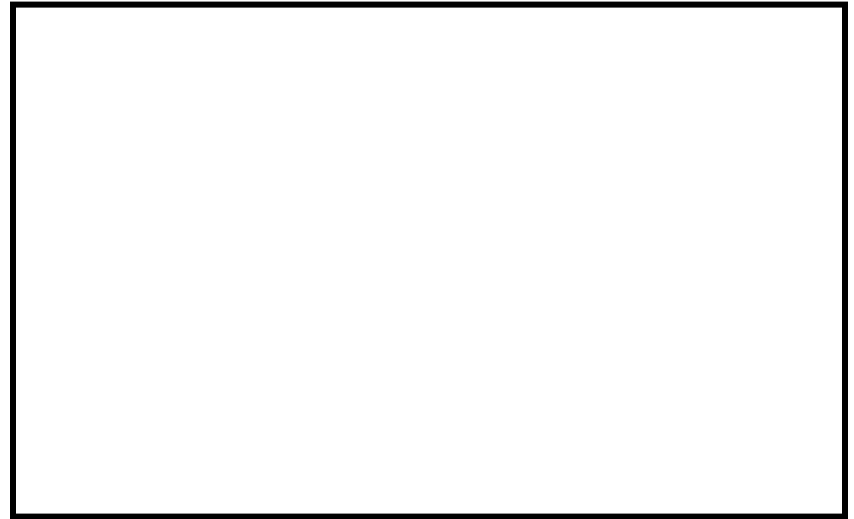


Laboratory ASM Worksheet

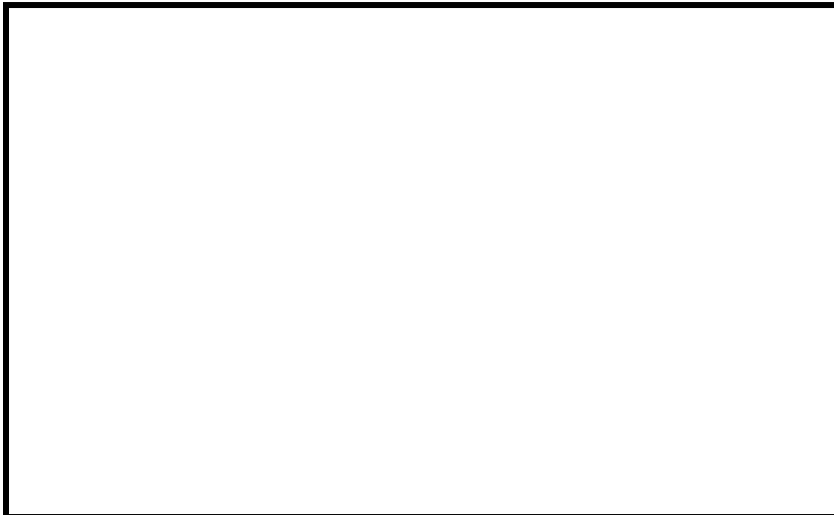
Thread_1



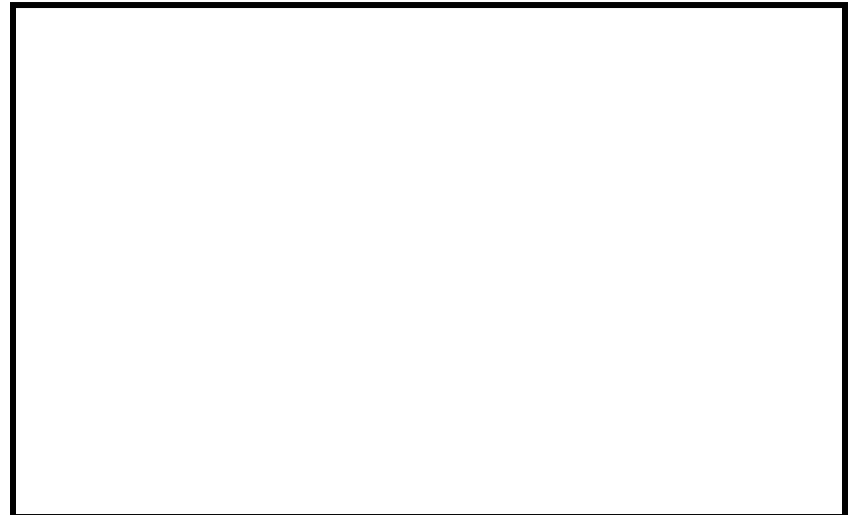
Thread_2



Thread_3



Thread_4



Use this worksheet to hand-draw your initial partitioned ASM design.

Show signal interconnection between the boxes.

Lab Exercise – Modulo Counter Design

- Three architectures – variations on a theme.
 - ✓ Fletcher’s basic Mod-10 4-bit Ripple counter [Fletcher, 1980].
 - ✓ A Mod-n 4-bit programmable Ripple counter – pass in the Mod value as an input, then use a separate control state machine to count to the mod value, then reset the main counter (like Fletcher’s technique, but more flexible).
 - ✓ A concurrent Mod-n 4-bit Ripple counter – have separate state machines for each state register, have one control the clock pulse of the next, which controls the clock of the next, etc. This more clearly shows the “rippling” of the count through the design.
- Lab Assignment.
 - ✓ Create ASM models for each of these Ripple counter architectures in Nimbus. You will need to complete the designs (as these are abstract ideas I am giving you), define the buses, and create the various control structures between the ASM threads. Make sure you have inputs (where required) and counter outputs.
 - ✓ Synthesize each of these in Synopsys® (or other logic synthesis tool, if available).
 - ✓ Turn in output consisting of Nimbus models, Synopsys schematics, and reports.
 - ✓ Remember, when using Synopsys: (1) Read in the file `nimbus_bit.vhdl` and/or `nimbus_MVL9_Synopsys.vhdl` files first, (2) read in your .vhdl design file output from Nimbus, (3) Export to EDIF (a .edf file) after Implementing the circuit. Then run it in Xilinx and generate the PAR report.