EE273 Lecture 16
Asynchronous State Machines, Pipelines, and Iterative Circuits

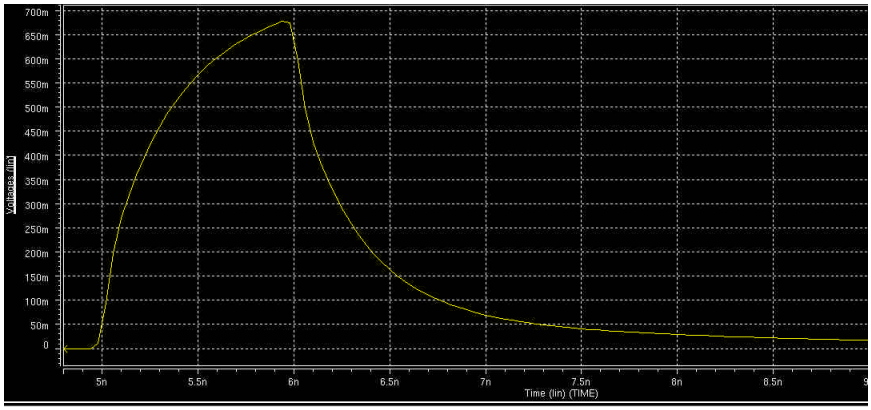
November 18, 1998

William J. Dally
Computer Systems Laboratory
Stanford University
billd@csl.stanford.edu

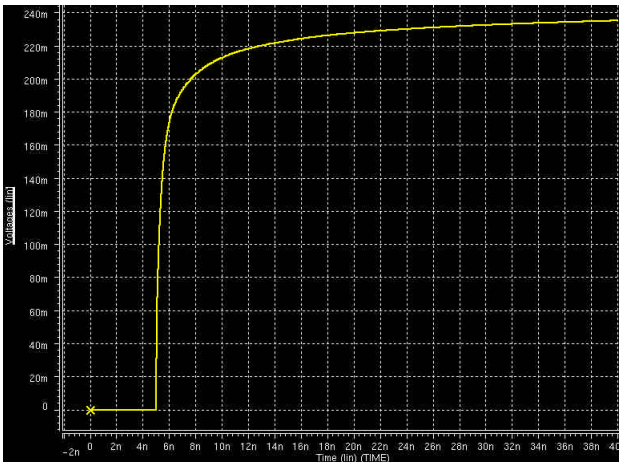
Today's Assignment

- Term Project
  - checkpoint 2 due on Wednesday 11/25
- Reading
  - Sections 5.1, 5.2, and 5.5

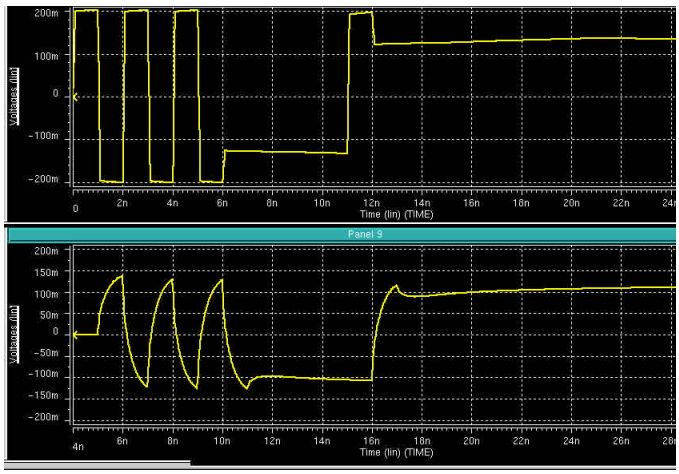
Some Comments about the Project



Project 2/3



Project 3/3



A Quick Overview

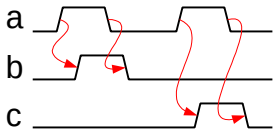
- Asynchronous State Machines
- Asynchronous Pipelines

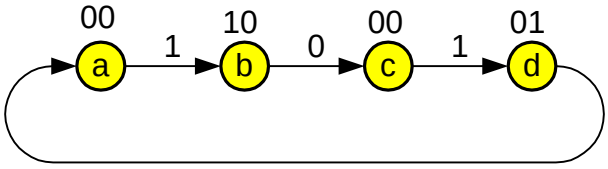
Asynchronous State Machines

- General case of sequential digital circuits
- Input transitions may cause change of
  - output values
  - internal state
- In the general case, several inputs may change simultaneously
  - not *fundamental mode*
  - multiple changes may represent *concurrency* or *choice*
- We can represent ASMs by
  - their input/output waveforms
  - a state table
    - also called a *flow table*
  - a state diagram

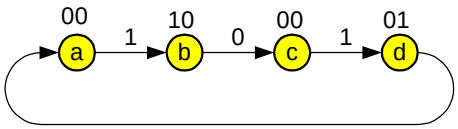
Example, a *Toggle* Circuit

- A toggle circuit
  - has one input, a, and two outputs, b and c.
  - a, b, and c are all RZ event-only signals (no value)
  - each odd event on a causes an event on b
  - each even event on a causes an event on c





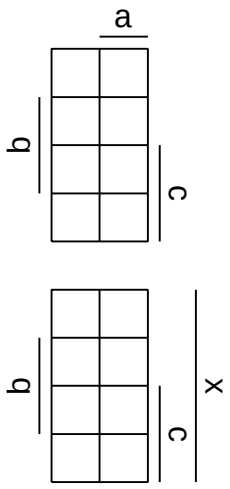
State Table Representation



		Next State	
State	Out	0	1
a	00	a	b
b	10	c	b
c	00	c	d
d	01	a	d

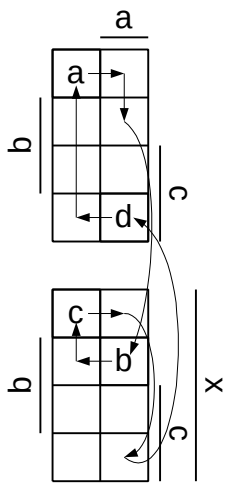
Trajectory Maps

- A Karnaugh map showing the *dynamic* behavior of the circuit
  - arrows indicate a *trajectory* through the state space
  - input transitions are *horizontal*
  - output and state variable changes are *vertical*
- Insert state variables as needed to keep distinct states separate
  - e.g., after a↑, b↑ in toggle circuit



Trajectory Map for the Toggle Circuit

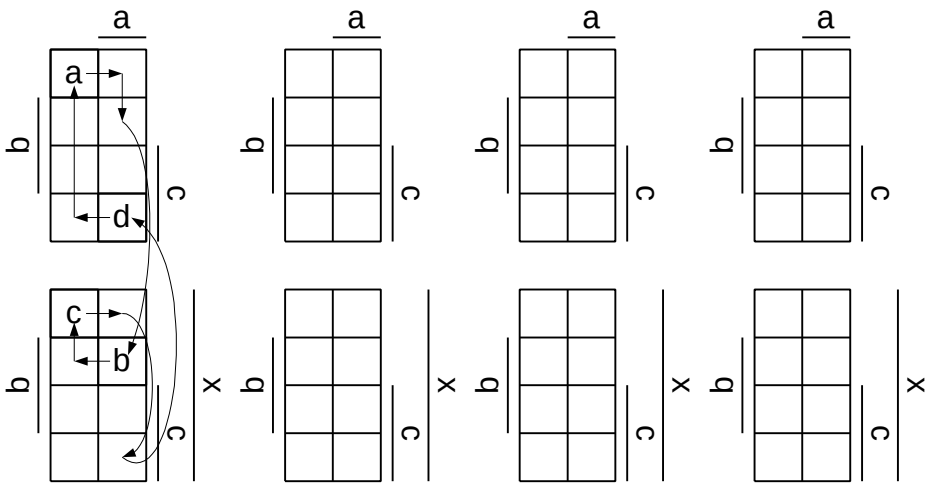
- Each arrow denotes a state transition
- Stable states are denoted with letters
- This is not the only possible sequence. Can you suggest another one?



From Trajectory Map to K-Map

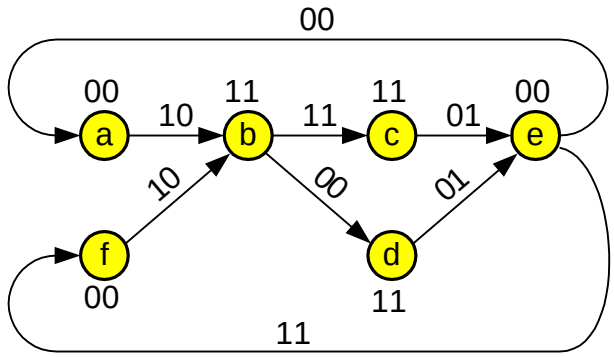
- For the K-map for output or state variable z
  - if an arrow leads to a state with z=1, mark the state where the arrow starts to 1
  - if an arrow leads to a state with z=0, mark the state where the arrow starts to 0
- Cover any hazards along the trajectory when selecting implicants to cover the logic function
- No need to cover off-trajectory hazards

Logic Design for the Toggle Circuit From Trajectory Map to K-Map



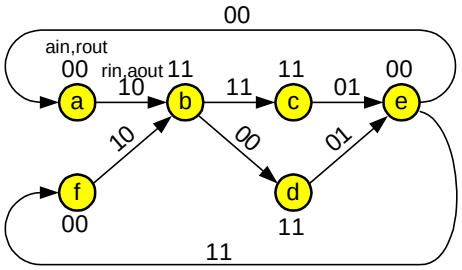
Another Example, A FIFO Control Element

- Arrows labeled with inputs
  - rin, aout
- States labeled with outputs
  - ain, rout



Concurrency and Choice

- In states b and e both inputs can change
- Does it matter which changes first?
- If they both change at the same time, what does the circuit do?
- This is an example of *concurrency*
  - move in one step from b to e
  - also called a non-critical race

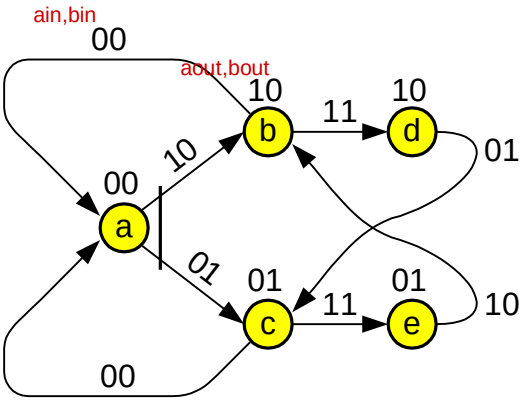


EE273, L16, Nov 18, 1998

15

An Arbiter

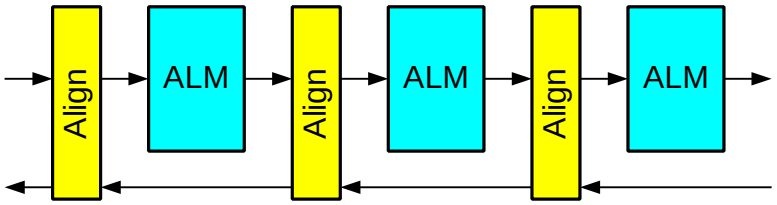
- In state a, both ain and bin can change
- Here it matters which one changes first
- Here the simultaneity represents *choice,* not *concurrency*
- In states b and c the simultaneous input changes represent *concurrency*
- Where there is choice, we need synchronization (an arbiter)



EE273, L16, Nov 18, 1998

16

A Simple Asynchronous Pipeline



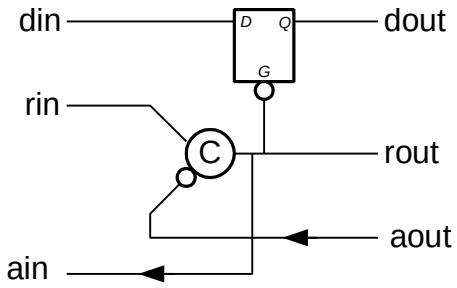
- Align blocks separate async logic modules
  - wait for all inputs valid, then signal previous align block
  - when signal received from forward align block, set inputs invalid (if RZ) and allow next set of inputs to enter

Asynchronous Pipelines

- Very different from synchronous pipelines
- Delay of pipe is sum of delays of stages
  - not rounded up to delay of longest stage
- The number of problems in the pipe is flexible (≤ the number of stages
  - Different branches of a pipe need not have the same number of stages
  - The first problem may exit before the second problem enters regardless of the number of stages

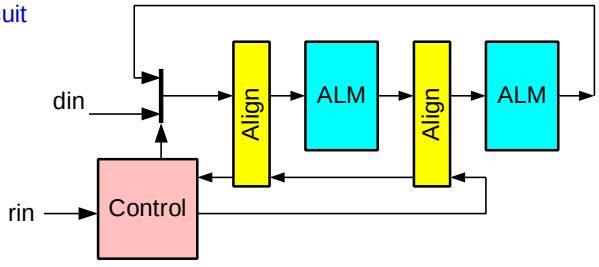
A Simple Align Block

- What does the state diagram of this circuit look like?
- What is wrong with this simple approach?
- How can we fix it?



Asynchronous Iterative Circuits

- Can we feed the output of an asynchronous pipeline back to its input?
- How does such a circuit behave?



Next Time

- Power Distribution