CS221: Digital Design

http://jatinga.iitg.ernet.in/~asahu/cs221

FSMD and ASM

A. Sahu

Dept of Comp. Sc. & Engg.

Indian Institute of Technology Guwahati

ASM Overview

- Drawbacks of state diagrams for real systems:
  - Many inputs & many outputs -> awkward to list all of these as each transition arc.
  - On any given arc
    - Typically most inputs are don't care
    - Typically most outputs are unchanged from the settings in the previous state
    - Tedious & repetitive to list exhaustively

ASM Overview

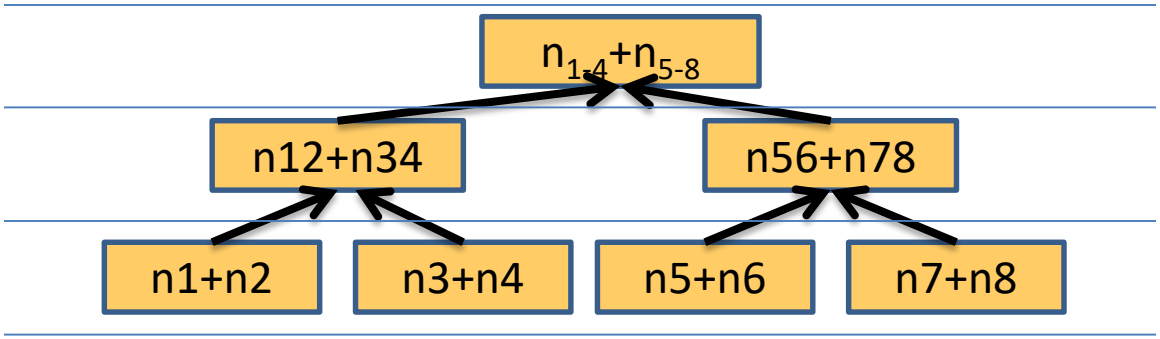
- Not a clear structure for illustrating/designing control flow
- What about generic memory/data
  - Do they really need to be part of the state? If we have many bits of data, this would lead to a huge state
  - **E.g. state diagram for counter or shift register is pointless**
    - **32 bit counter have 2^{32} states**

ASM Overview

- Some problems analogous to before
  - Combinational:
    - Small problems – truth tables ok/easy
    - Adders, Muxes – TT get out of hand
      - **Design 2 level ckt for 32 Adders : 64 inputs and 32 output using TT method, worst case delay (2^{63} or gate)**
      - **Worst case OR gate size or # of Product term ==> $2^{N-1}-1$**
      - **CLA : prefix sum example**

CLA: PrefixSum

- Sum of 8 numbers
  - Serial Require 1 adder, Delay n
    (n1+n2+n3+n4+n5+n6+n7+n8)
  - Parallel, require n/2 adder, Delay lg n



- Prefix Sum of 8 numbers
  - 8 numbers, at each position sum upto that number
  - n1, n1+n2, n1+n2+n3, …., n1+n2+..+n8
  - Two pass tree operation : log N time

ASM Overview

- Some problems analogous to before
  - Sequential:
    - Small – state diagrams easy
    - Real, Data – state diagrams not helpful

Algorithmic State Machine

Algorithmic State Machine –

representation of a Finite State Machine

suitable for FSMs with a larger number of inputs and outputs compared to FSMs expressed using state diagrams and state tables.

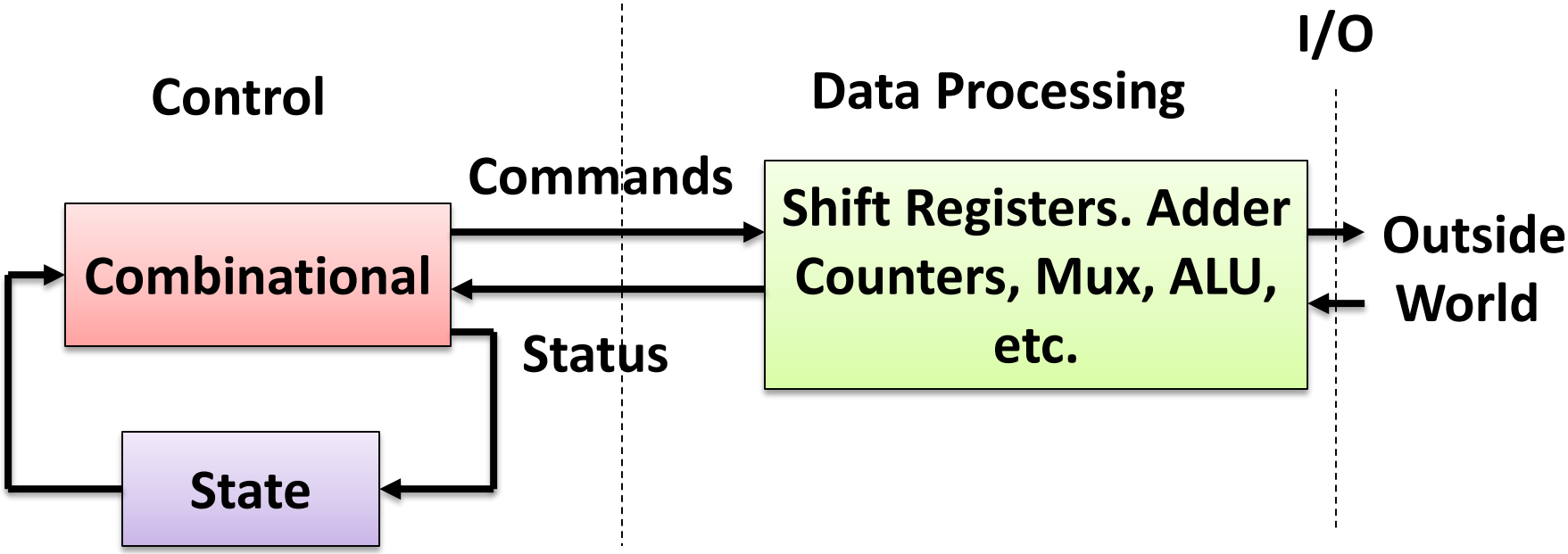
ASM Overview

- We need to separate controller & data processor
  - Controller – What actions need to be taken? What is fundamental operating mode?
  - Processor – Undertake the action. Manipulate the data

The ultimate Goal of this course : Design using Control Path + Data Approach : RTL Design

ASM Overview

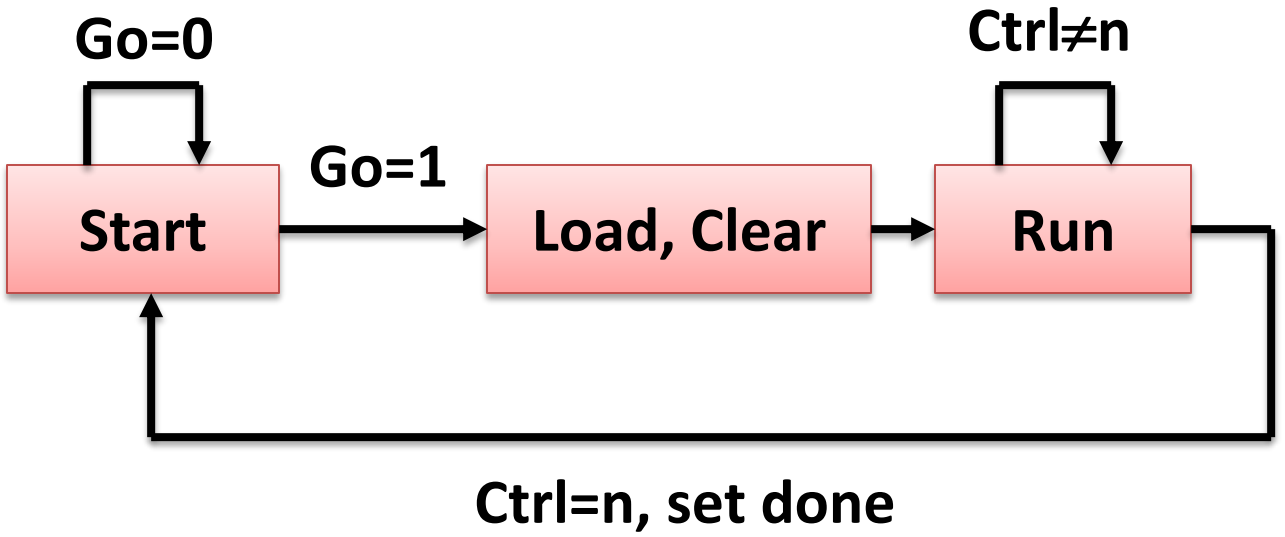
- Control and data path interaction



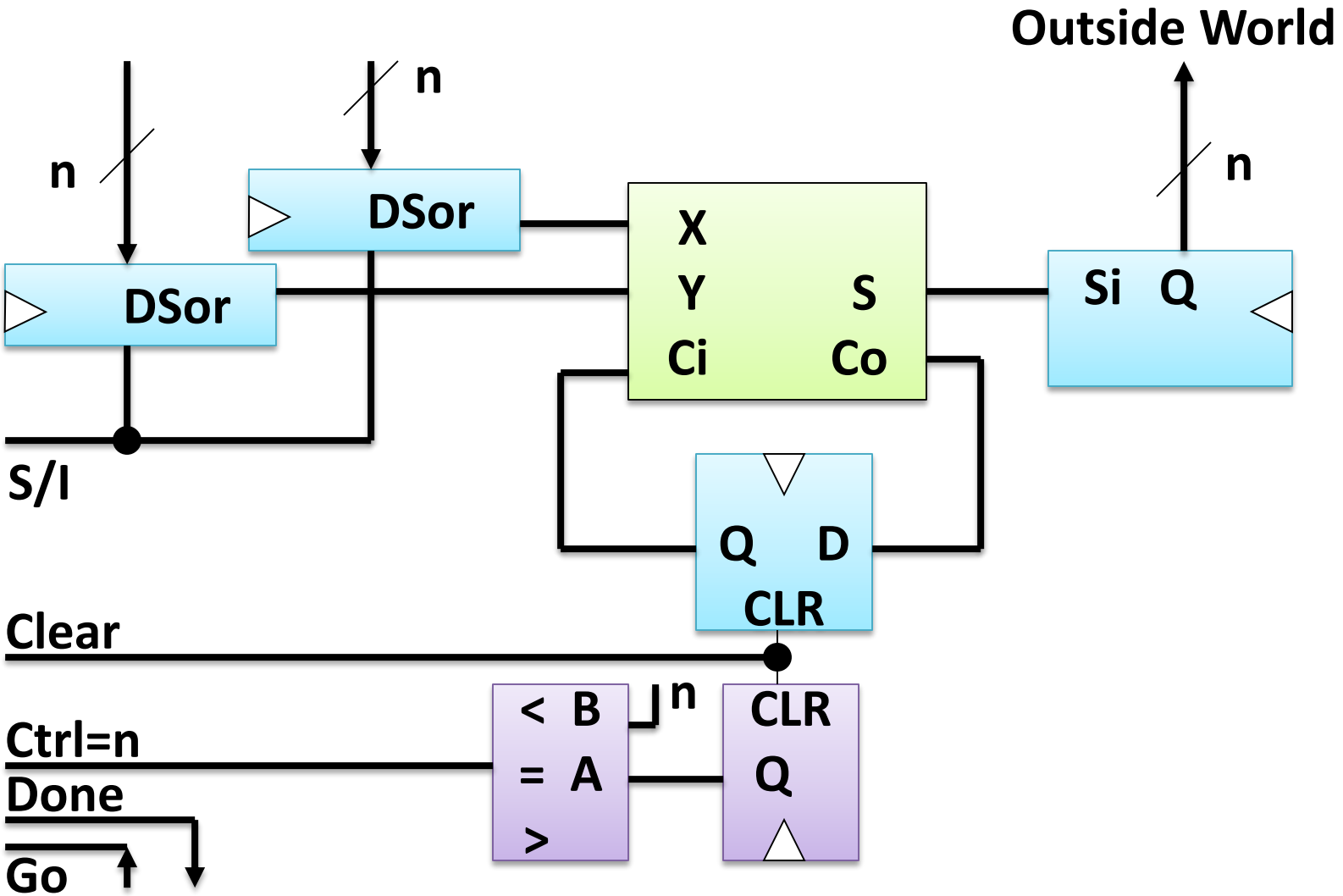
- Our circuit is now explicitly separated

ASM Overview : High Level

- Ex. Serial Addition
- Control Part/Path



Serial Addition Data Path



ASM Design : Data processing

- What sorts of manipulations of the input and output data are requested?
- How many/what sorts of things need to be stored?
- How to design
  - Ad hoc/creative/by insight
  - List requested operations/manipulations
  - Include initialization controls
  - Include status lines

ASM Design : Control logic

- All of the commands to the data proc. logic need to be controlled,
- And the status lines need to be monitored and acted upon.
- ASM charts are like state diagrams, but without specific drawbacks.
  - Don’t list all inputs for each transition – don’t care inputs
  - Don’t list all outputs for each state – not changed outputs

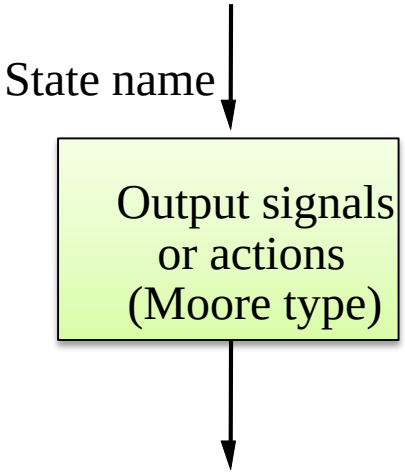
ASM Design

- How to design - ASM chart/state diagram (**for small problems**)
  - State assignment
  - State table
  - Kmap-gates/FF/Reg Mux Dec/EPROM, or, creatively, a combination of them

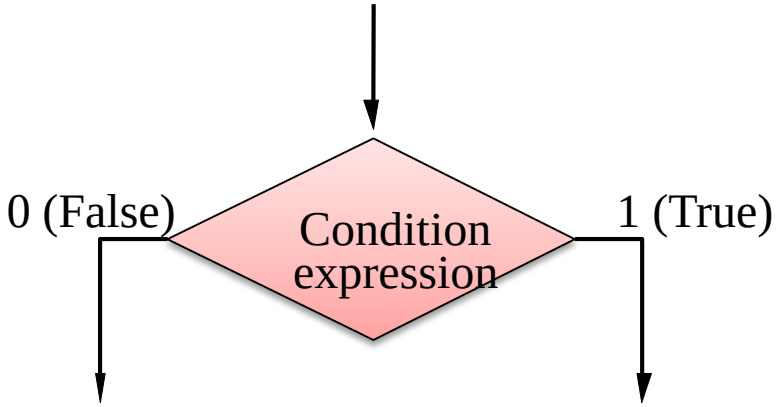
ASM Design

- ASM charts are like flowcharts, with a few crucial differences.
- Be careful, especially with timing.
- Three type components/Box
  - **State Box**
  - **Decision Box**
  - Combinational Box/Transition Box/**Conditional Box**

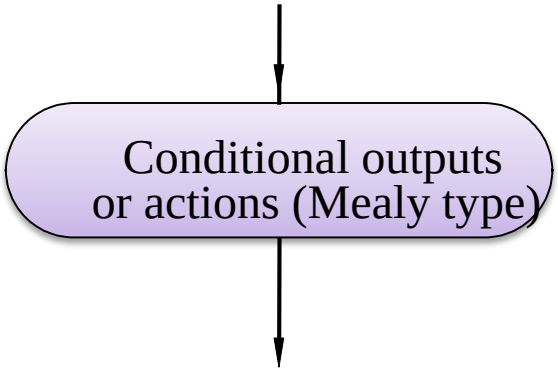
Elements used in ASM charts



(a) State box



(b) Decision box



(c) Conditional output box