Conclusion

- What we did the last 10 weeks
- What you should now know
- The relationship of CSE370 to other courses in CSE core requirements
- Follow-on courses
- Evaluation

What we did . . .

- Basics of logic design
  - combinational and sequential circuits
- Design methodologies
  - finite-state machines and their implementation
- Modern specification methods
  - schematics and hardware description languages
- Modern set of CAD tools
  - DesignWorks and VerilogModeler
- Differences and similarities in hardware and software design
  - inherent parallel nature of hardware
  - input/output/state encoding

You should now . . .

- Feel confident that you can design a circuit to perform virtually any function
  - come up with problem specification
  - separation of data path and control concerns
  - consider change to meet criteria of size/cost
- Have a basic understanding of the hardware costs of a particular function
  - variable <–> register/memory
  - assignment <–> register load/memory write
  - conditional <–> logic equation
  - program execution <–> sequence of states
  - control flow <–> state machine
- Have a general understanding of how computers are organized
  - typical instruction cycle
  - control/data-path/memory
  - finite-state controller

CSE370’s place in CS&E core curriculum

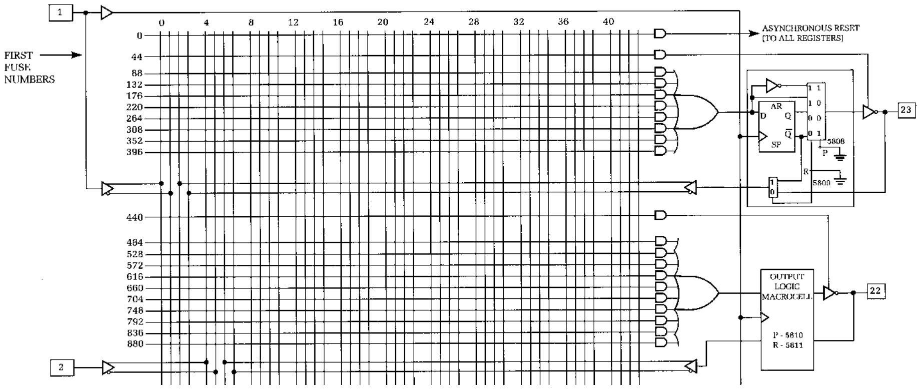
- CSE321 – Discrete Mathematics
  - basic mathematics used in computing
- CSE322 – Introduction to Formal Methods in Computer Science
  - finite automata, languages, computability
- CSE326 – Data Structures
  - fundamental elements of data organization and algorithms
- CSE341 – Programming Languages
  - survey of the motivation/advantages/disadvantages of several languages
- CSE370 – Introduction to Digital Logic
  - computation in hardware, parallelism, and computer structure
- CSE378 – Machine Organization and Assembly Language
  - basic architecture and mapping of software to machine instructions

Courses that have CSE370 as a pre-requisite

- CSE467 – Advanced Digital Design (required for CompE)
  - construction and debugging techniques, design labs
  - complex PLDs, field-programmable gate arrays (FPGAs)
  - behavioral synthesis and underlying algorithms
- CSE468 – VLSI Design (elective)
  - custom and semi-custom integrated circuit design
  - realization of logic elements with transistors
- CSE471 – Computer Design
  - high-level simulation of processor architecture (in Verilog)
  - architecture tradeoffs and evaluation (pipelining, superscalar, etc.)
- CSE477 – Digital System Design
  - embedded processors
  - large team design project

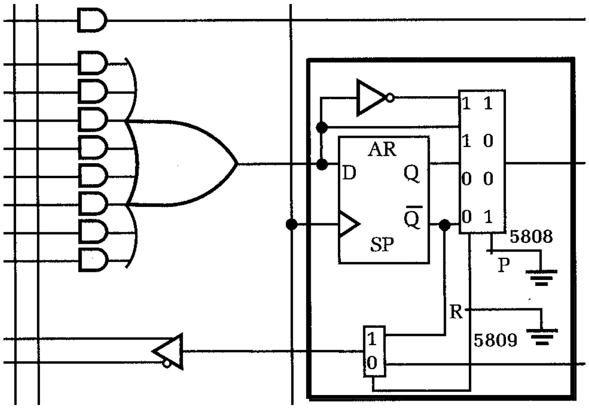
Programmable logic devices

- Programmable array logic with sequential logic element
  - output can be combinational or sequential, inverted, tri-stated
  - output circulated back in as input to array



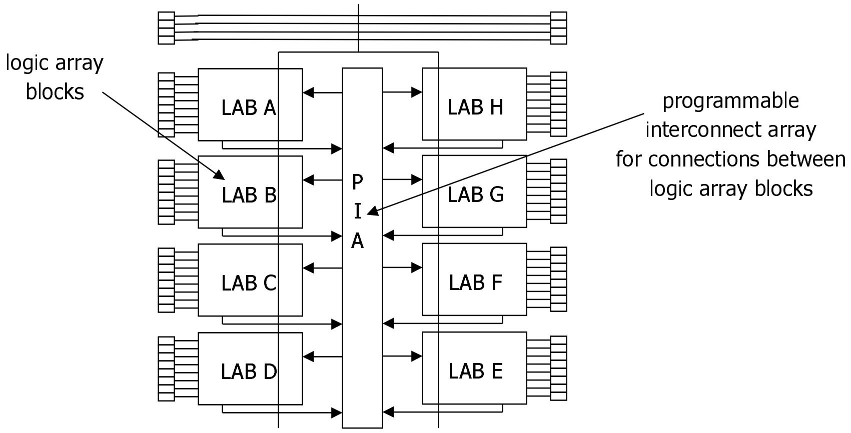
Programmable logic devices (cont'd)

- I/O cell for each PAL output
  - provides flexibility on output
  - use of pin as input (tristate capability)
  - reset/preset on flip-flop



Complex PLDs (CPLDs)

- Multiple PLDs in one package
- Programmable interconnect

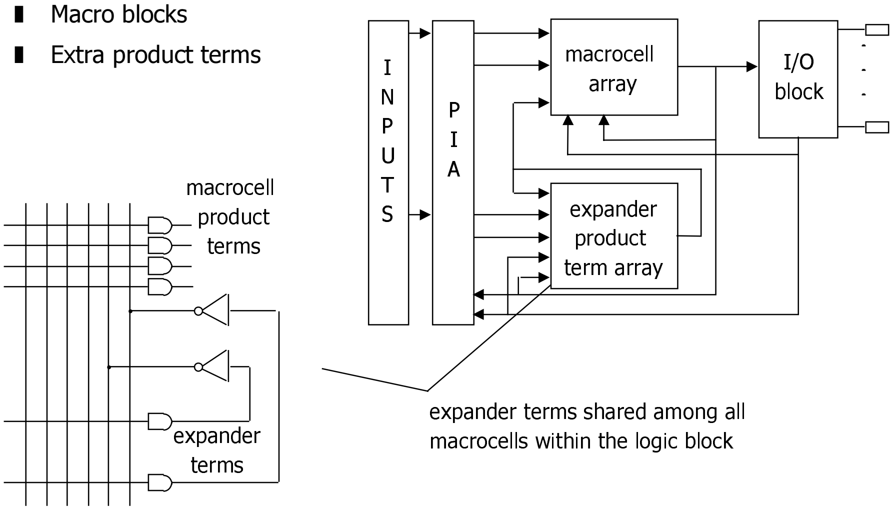


logic array blocks

programmable interconnect array for connections between logic array blocks

Complex PLSs (CPLDs) (cont'd)

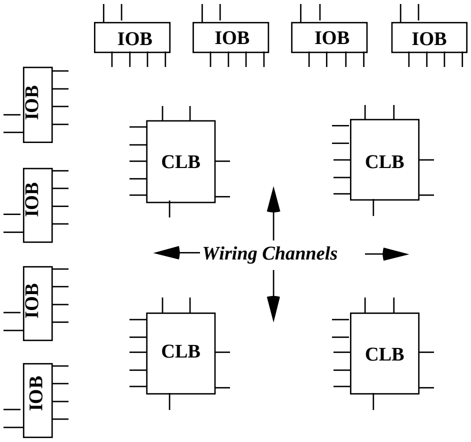
- Macro blocks
- Extra product terms



expander terms shared among all macrocells within the logic block

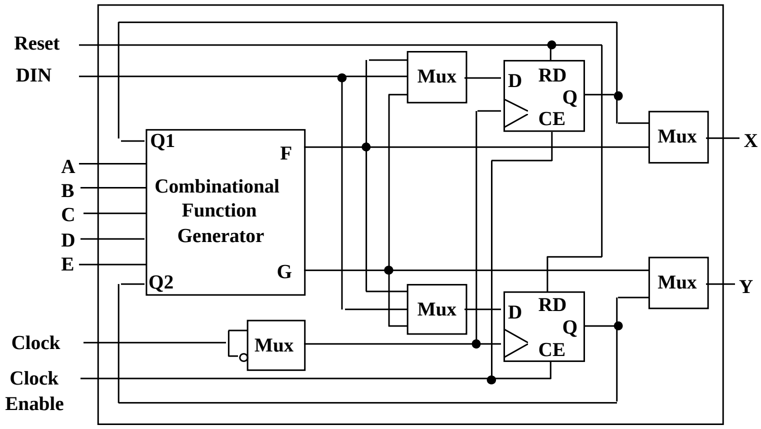
Field-programmable gate arrays (FPGAs)

- Array of logic cells implemented as lookup tables – any function of 5 inputs
- Programmable interconnect - different length wires connecting cells
- Programmable I/O cells



Field-programmable gate arrays (FPGAs) (cont'd)

- Programmable logic cells
  - combinational logic function
  - 2 flip-flops



Embedded processors

- Ever faster processors
  - software doing the job hardware used to do
- Specialized I/O capabilities
  - controllability and observability of individual wires (I/O ports)
- Timer and communication co-processors
  - serial line interface
  - timers
  - input capture (record time of event)
  - output compare (generate at a specified time)

Summary

- What the entire course was about
  - converting solutions to problems into combinational and sequential networks effectively organizing the design hierarchically
  - doing so with design tools
    that lets us handle designs effectively and efficiently
  - taking advantage of optimization opportunities
    (don't cares, encodings, etc.)
  - appreciating the inherent parallel nature of hardware
- That took ten weeks
  - but there is a lot more to it
  - this was just the beginning
  - hope to see you in 467 or 477 or . . .

Evaluation

- These really do matter
  - department takes them very seriously
  - crucial feedback for instructor
- Take the time and fill it out completely
  - additional questions can be answered on back of form
  - yellow sheets are the most valuable – write down your thoughts