LECTURE 3
PIPELINING

B649

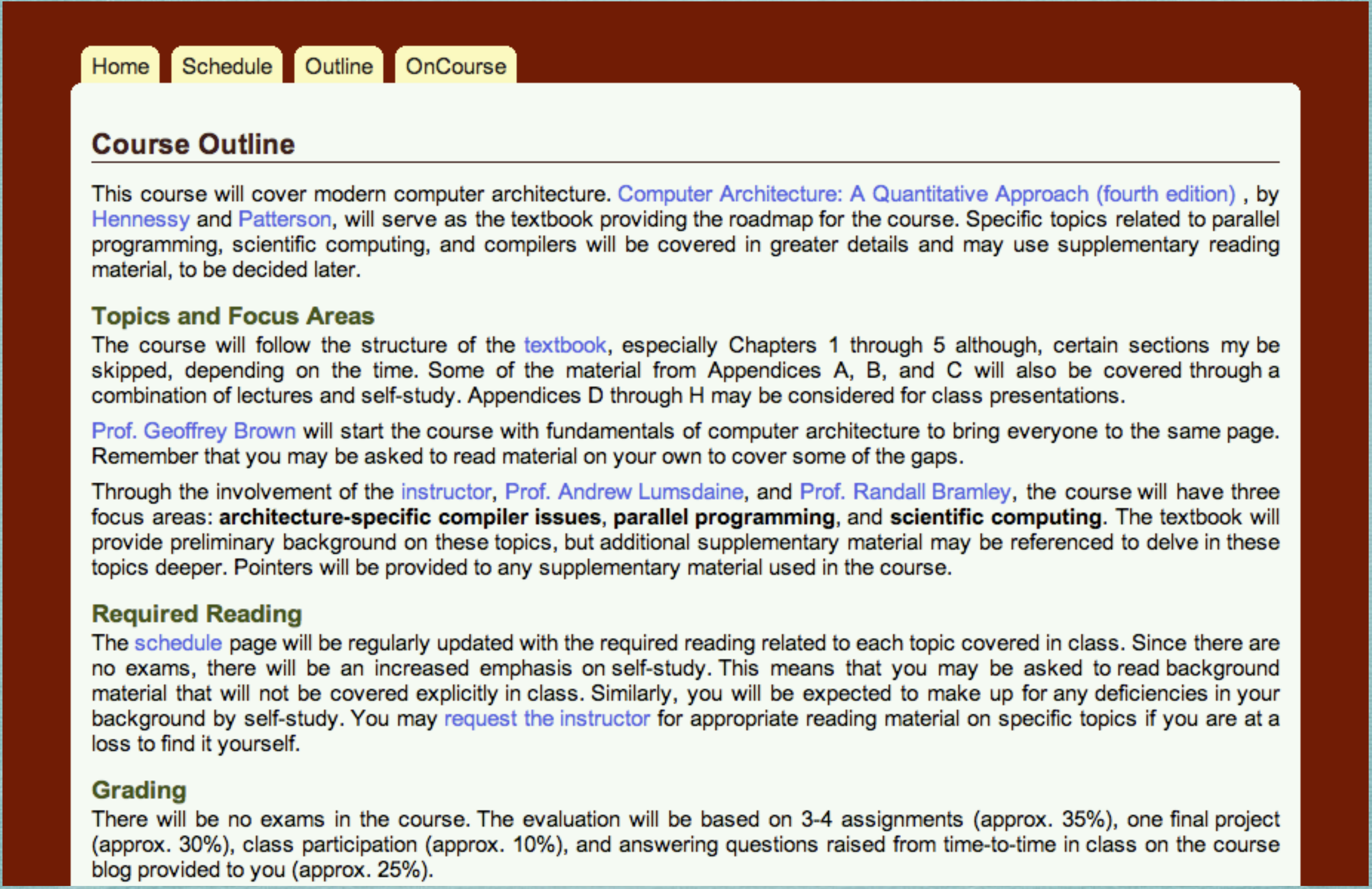
Parallel Architectures and Programming

“First” Lecture

- Introductions
- Course plan
  - ★ Administrative details
  - ★ Resources
  - ★ Grading
- Pipelining

INTRODUCTIONS

Course Plan



Course Outline

This course will cover modern computer architecture. Computer Architecture: A Quantitative Approach (fourth edition) , by Hennessy and Patterson, will serve as the textbook providing the roadmap for the course. Specific topics related to parallel programming, scientific computing, and compilers will be covered in greater details and may use supplementary reading material, to be decided later.

Topics and Focus Areas

The course will follow the structure of the textbook, especially Chapters 1 through 5 although, certain sections my be skipped, depending on the time. Some of the material from Appendices A, B, and C will also be covered through a combination of lectures and self-study. Appendices D through H may be considered for class presentations.

Prof. Geoffrey Brown will start the course with fundamentals of computer architecture to bring everyone to the same page. Remember that you may be asked to read material on your own to cover some of the gaps.

Through the involvement of the instructor, Prof. Andrew Lumsdaine, and Prof. Randall Bramley, the course will have three focus areas: **architecture-specific compiler issues**, **parallel programming**, and **scientific computing**. The textbook will provide preliminary background on these topics, but additional supplementary material may be referenced to delve in these topics deeper. Pointers will be provided to any supplementary material used in the course.

Required Reading

The schedule page will be regularly updated with the required reading related to each topic covered in class. Since there are no exams, there will be an increased emphasis on self-study. This means that you may be asked to read background material that will not be covered explicitly in class. Similarly, you will be expected to make up for any deficiencies in your background by self-study. You may request the instructor for appropriate reading material on specific topics if you are at a loss to find it yourself.

Grading

There will be no exams in the course. The evaluation will be based on 3-4 assignments (approx. 35%), one final project (approx. 30%), class participation (approx. 10%), and answering questions raised from time-to-time in class on the course blog provided to you (approx. 25%).

Course Plan: Administrative Details

- Use OnCourse
  - grades
  - group e-mail
- Visit schedule page
  - keep up with required reading
- Maintain blogs
  - answer questions
  - post interesting links
  - post comments (use as discussion forum)

Course Plan: Resources

- Textbook
  - ★ case studies
  - ★ accompanying CD
- Lecture notes
  - ★ slides
- Internet
  - ★ class blogs
- Supplementary reading

Course Plan: Grading

- No exams
- Participation
  - in class
  - presentation(s)
    - Appendices D through H, assembly programming, GPUs
- Blog questions
- Assignments
  - 3 to 4
- Project
  - topics: parallel architecture and/or programming

Disclaimer

Disclaimer

I am **not** a computer architect.

Disclaimer

I am **not** a computer architect.

I am not even a hardware designer!

Disclaimer

I am **not** a computer architect.

I am not even a hardware designer!

I **am** an Electrical Engineer.

Disclaimer

I am **not** a computer architect.

I am not even a hardware designer!

I **am** an Electrical Engineer.

Instructor learns as much in a seminar course as students.

DIGITAL HARDWARE DESIGN

Digital Hardware Design in a Slide

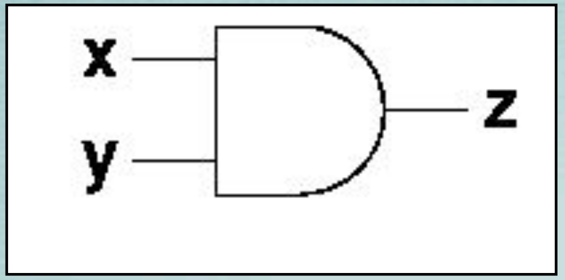
- Basic building blocks
  - transistors
  - gates
  - latches
  - flip-flops
- Circuits
  - combinatorial
  - sequential
    - asynchronous
    - synchronous (clocked)

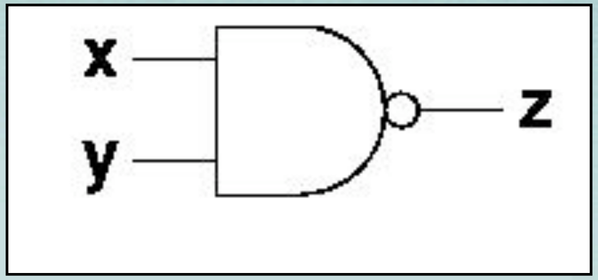
Digital Hardware Design in a Slide

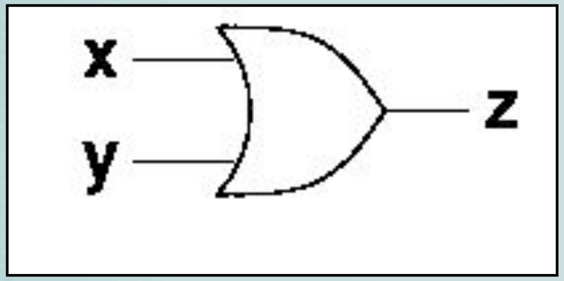
- Basic building blocks
  - transistors
  - gates
  - latches
  - flip-flops
- Circuits
  - combinatorial
  - sequential
    - asynchronous
    - synchronous (clocked)

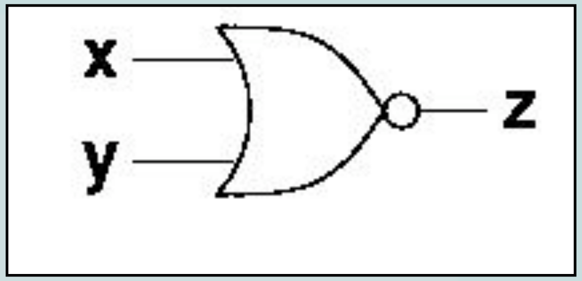
Reference: Digital hardware basics

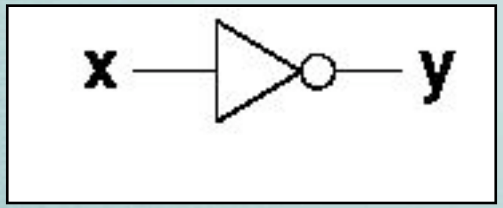
Gates

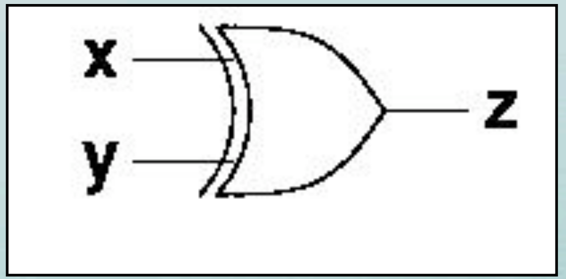




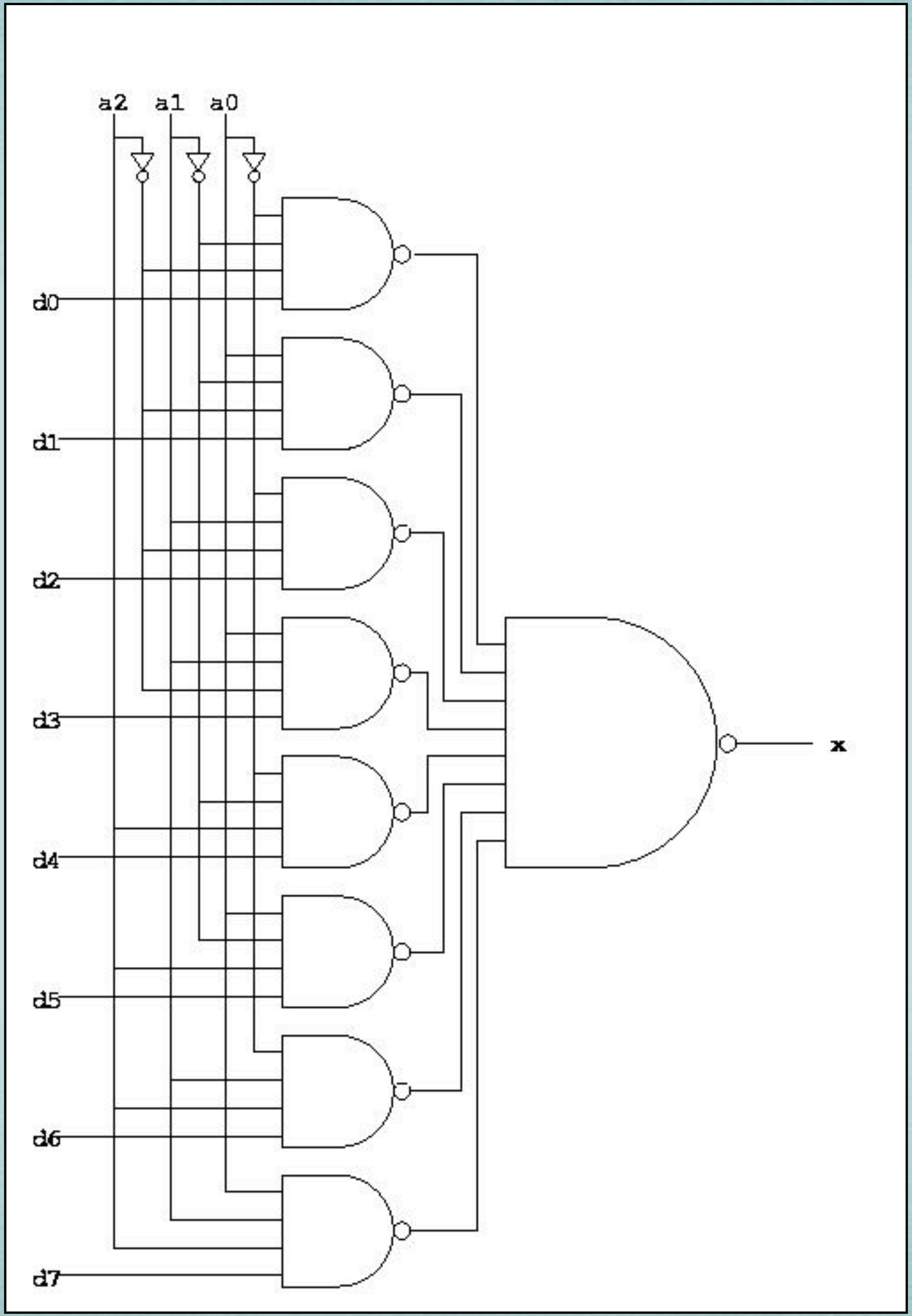




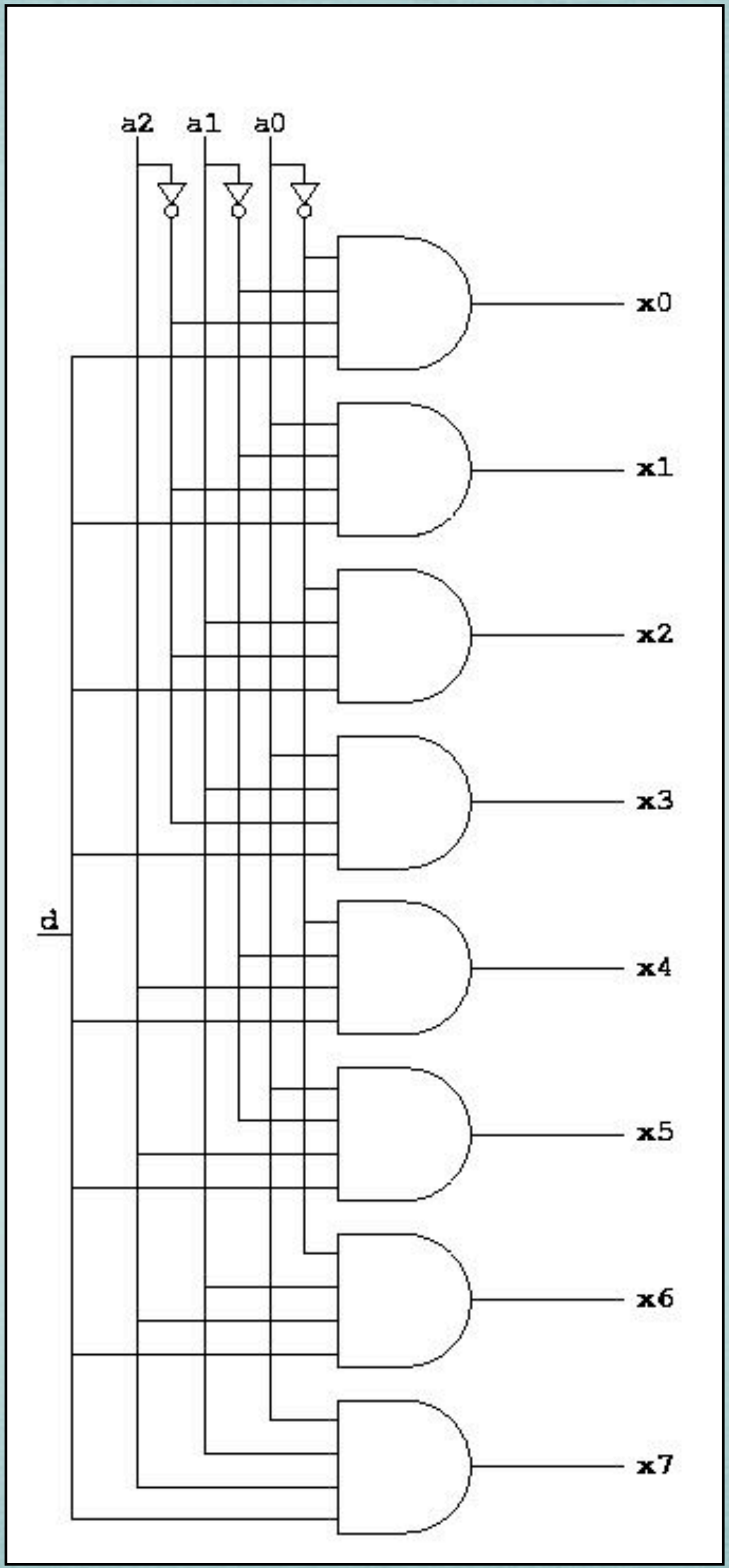




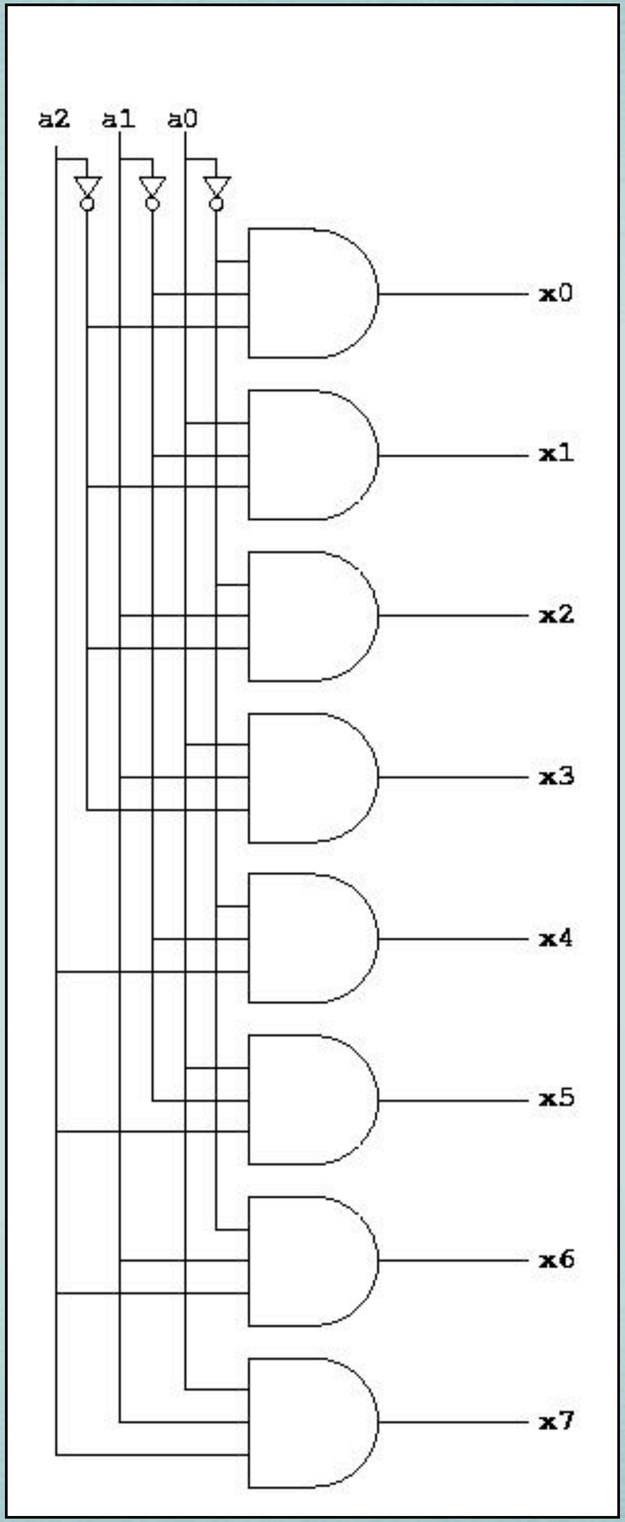
Multiplexer



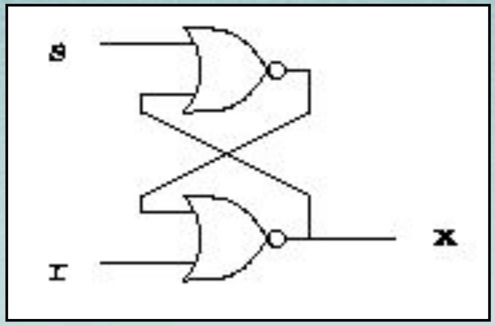
Demultiplexer

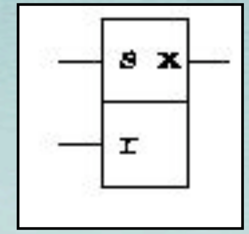


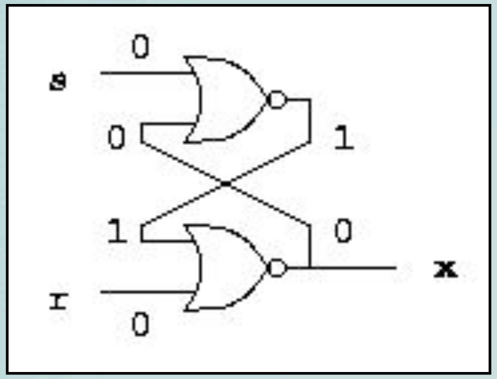
Decoder

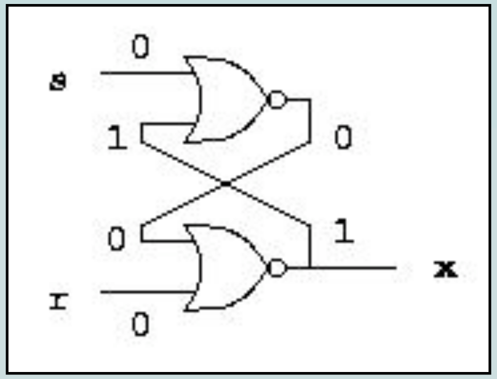


SR Latch

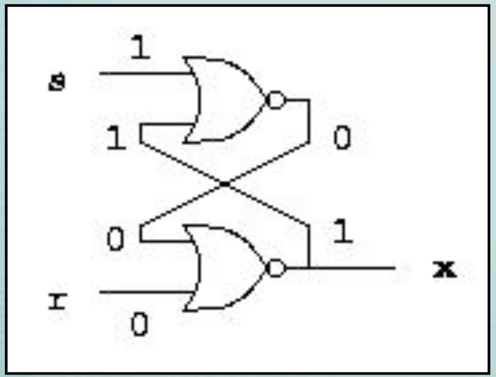


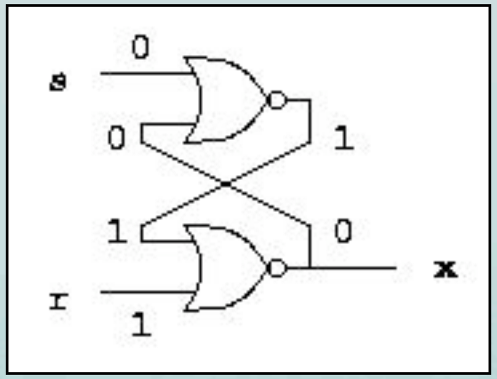






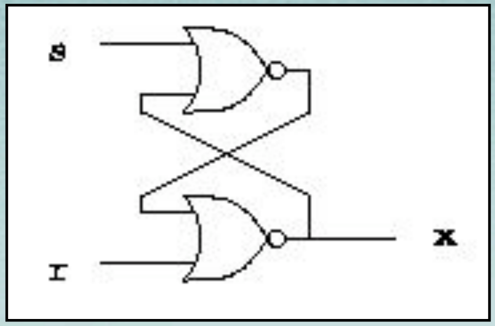
At most one input is 1

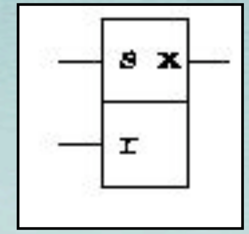


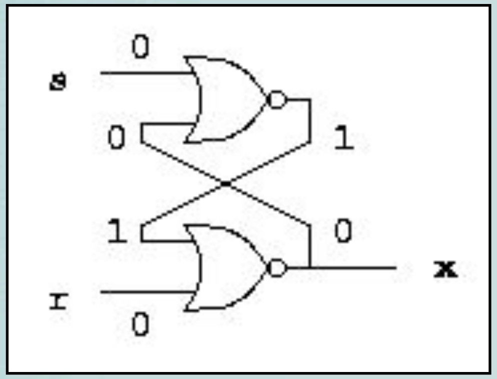


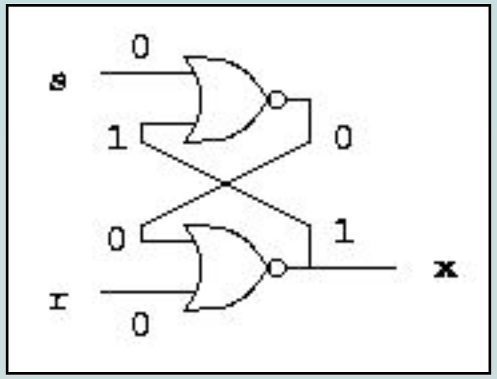
Latch “remembers”

SR Latch

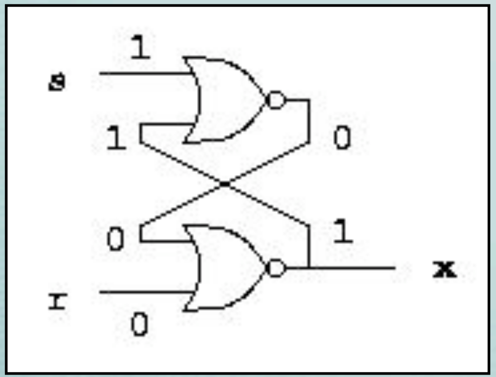


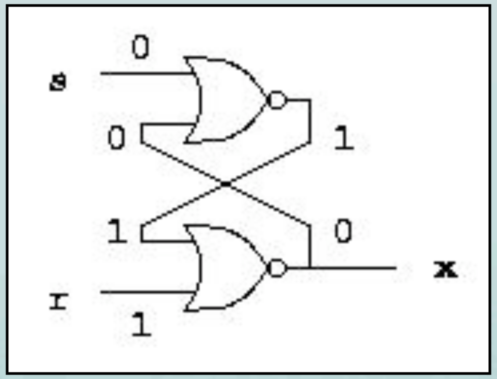






At most one input is 1

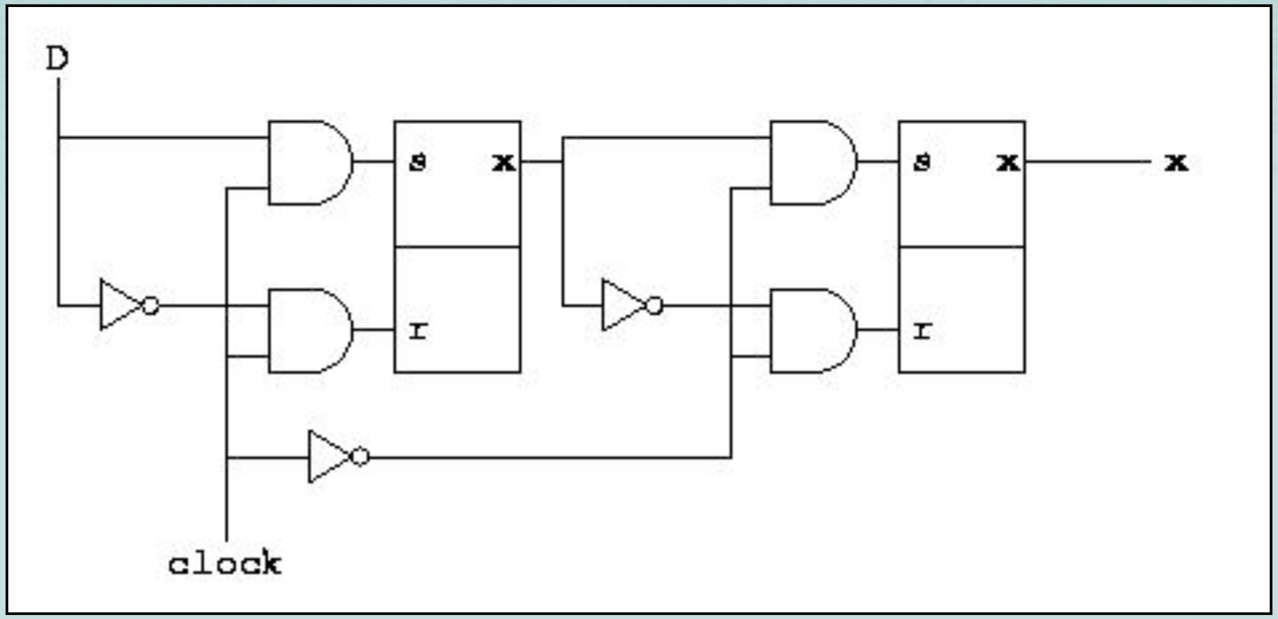


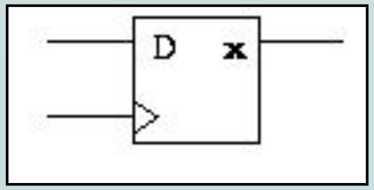


Latch “remembers”

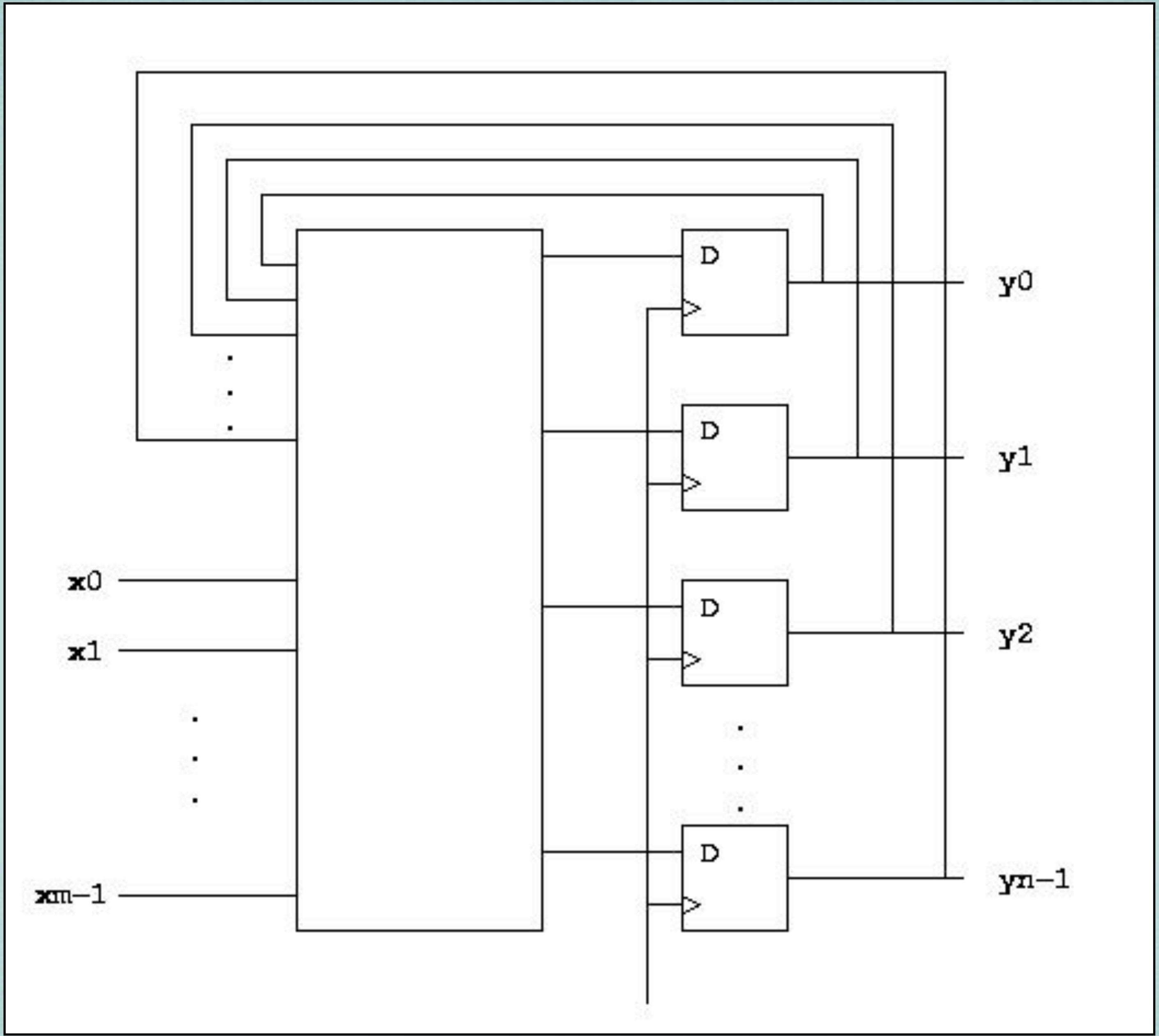
What if both the inputs are 1?

Flip-flops

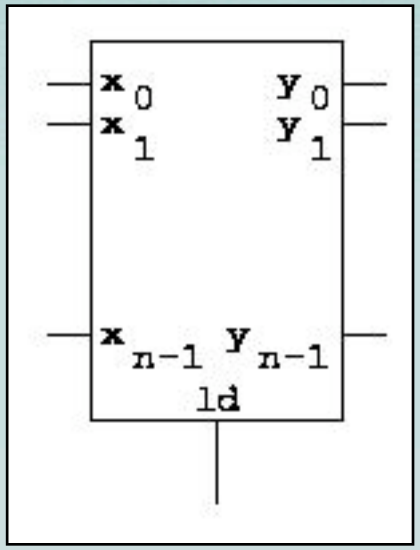




Sequential Circuits



Registers



ld	x3	x2	x1	x0	y3	y2	y1	y0	y3'	y2'	y1'	y0'
0	–	–	–	–	c3	c2	c1	c0	c3	c2	c1	c0
1	c3	c2	c1	c0	–	–	–	–	c3	c2	c1	c0

Computer Clocks

- Computer hardware as clocked sequential circuit
- Clock speed decided by
    - speed of the latches
    - speed of the combinatorial circuit
    - propagation delay
    - interconnects
        - impedance
        - power consumption
        - clock skew
    - power consumption
    - cost