INTRODUCTION

B649

Parallel Architectures and Programming

Outline

- Introductions
- Course plan
  - ★Administrative details
  - ★Resources
  - ★Grading
- Digital hardware basics

INTRODUCTIONS

Course Plan

Home | Schedule | Outline | Guidelines | Blogs | OnCourse

Quick facts

Course	B649—Parallel Architectures and Programming (Topics in Computer Architecture)
Credits	3
Location	LH 019 (Lindley Hall)
Times	Mon & Wed, 4:00-5:15 PM
Instructor	Arun Chauhan
Office Hours	By appointment
Textbook	Computer Architecture: A Quantitative Approach (fourth edition), by John L. Hennessy and David A. Pattterson

Motivation

In 1997, IEEE Computer Society published a special issue on what could be achieved with a billion transistors on a single chip. Leading computer architects proposed seven different ideas. A subsequently published article in 2004 noted that none of those architectures has become mainstream. Such is the nature of research! However, a billion transistors on a chip is now a reality. A symmetric multi-core architecture is just one of the uses for billions of transistors on a chip. What other possibilities are there? Do these entail entirely novel ways of programming them? What are the consequences for application programmers? For compilers? This course will attempt to answer these questions by drawing on experts in the fields of architecture, parallel programming, high-performance applications, and compilers.

About the course

The course will cover modern computer architectures, including multi-cores and GPUs. There might be a few guest lectures. The rest will be divided between the instructor and student presentations. The course content will be based on a textbook (Computer Architecture, A Quantitative Approach (Fourth Edition)) and published technical papers and / or other supplementary material (to be decided).

Goals

This course aims to provide:

- A basic understanding of the architecture of modern machines;
- Programming and compiler challenges thrown by the modern architectures;
- Techniques currently being used, and researched, to efficiently and effectively program modern machines.

Prerequisites

A substantial knowledge of computer architecture is not assumed, although a fundamental understanding of digital hardware and computer architecture is expected. Experience with system-level programming will help. Familiarity with Unix / Linux is highly recommended.

Course-load

The course will involve 3-4 assignments and one final project. There will be no exams. Apart from the assignments and project, the evaluation will be based on class presentation, class participation, and answers to semi-open-ended questions that will be posed in class from time to time to be researched and answered on a course blog.

Course Plan

Home | Schedule | Outline | Guidelines | Blogs | OnCourse

Quick facts

Course	B649—Parallel Architectures and Programming (Topics in Computer Architecture)
Credits	3
Location	LH 019 (Lindley Hall)
Times	Mon & Wed, 4:00-5:15 PM
Instructor	Arun Chauhan
Office Hours	By appointment
Textbook	Computer Architecture: A Quantitative Approach (fourth edition), by John L. Hennessy and David A. Pattterson

Motivation

In 1997, IEEE Computer Society published a special issue on what could be achieved with a billion transistors on a single chip. Leading computer architects proposed seven different ideas. A subsequently published article in 2004 noted that none of those architectures has become mainstream. Such is the nature of research! However, a billion transistors on a chip is now a reality. A symmetric multi-core architecture is just one of the uses for billions of transistors on a chip. What other possibilities are there? Do these entail entirely novel ways of programming them? What are the consequences for application programmers? For compilers? This course will attempt to answer these questions by drawing on experts in the fields of architecture, parallel programming, high-performance applications, and compilers.

About the course

The course will cover modern computer architectures, including multi-cores and GPUs. There might be a few guest lectures. The rest will be divided between the instructor and student presentations. The course content will be based on a textbook (Computer Architecture, A Quantitative Approach (Fourth Edition)) and published technical papers and / or other supplementary material (to be decided).

Goals

This course aims to provide:

- A basic understanding of the architecture of modern machines;
- Programming and compiler challenges thrown by the modern architectures;
- Techniques currently being used, and researched, to efficiently and effectively program modern machines.

Prerequisites

A substantial knowledge of computer architecture is not assumed, although a fundamental understanding of digital hardware and computer architecture is expected. Experience with system-level programming will help. Familiarity with Unix / Linux is highly recommended.

Course-load

The course will involve 3-4 assignments and one final project. There will be no exams. Apart from the assignments and project, the evaluation will be based on class presentation, class participation, and answers to semi-open-ended questions that will be posed in class from time to time to be researched and answered on a course blog.

Quick Facts

Quick facts

Course	B649—Parallel Architectures and Programming (Topics in Computer Architecture)
Credits	3
Location	LH 019 (Lindley Hall)
Times	Mon & Wed, 4:00-5:15 PM
Instructor	Arun Chauhan
Office Hours	By appointment
Textbook	Computer Architecture: A Quantitative Approach (fourth edition), by John L. Hennessy and David A. Pattterson

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
- Subversion
  - ★assignments / projects
  - ★comments on your grades
- 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
  - ★options: term paper / programming / oral exam

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

B649: Parallel Architectures and Programming

Digital Hardware Design in a Slide

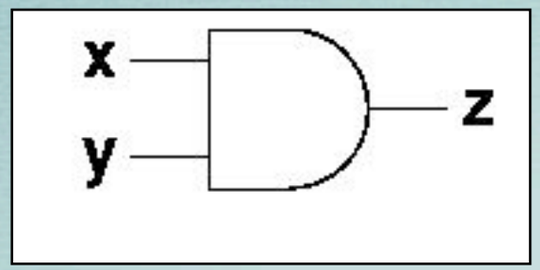
- Basic building blocks
  - ★transistors
  - ★gates
  - ★latches
  - ★flip-flops
- Circuits
  - ★combinatorial
  - ★sequential
    - ✳ asynchronous
    - ✳ synchronous (clocked)

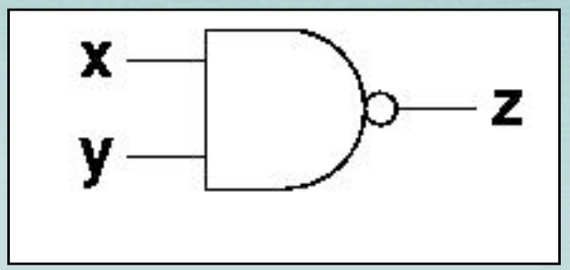
Digital Hardware Design in a Slide

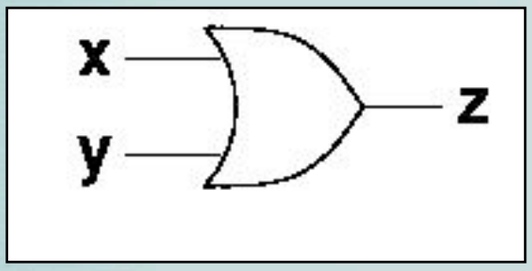
- Basic building blocks
  - ★transistors
  - ★gates
  - ★latches
  - ★flip-flops
- Circuits
  - ★combinatorial
  - ★sequential
    - ✳ asynchronous
    - ✳ synchronous (clocked)

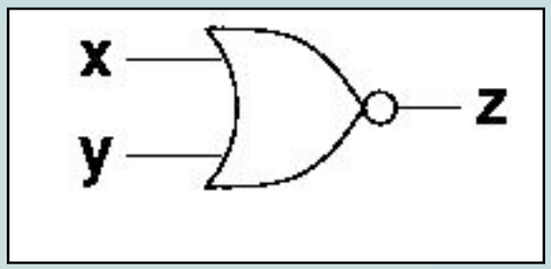
Reference: Digital hardware basics

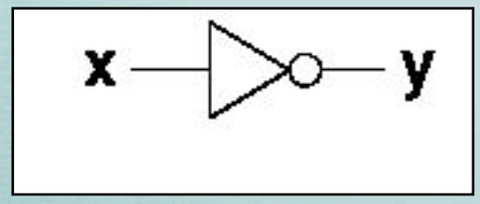
Gates

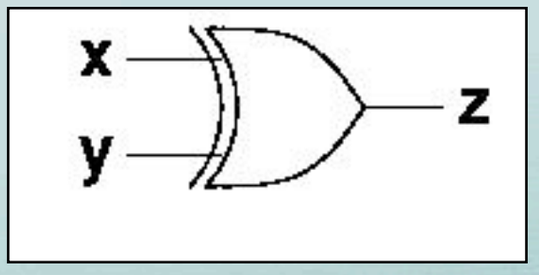




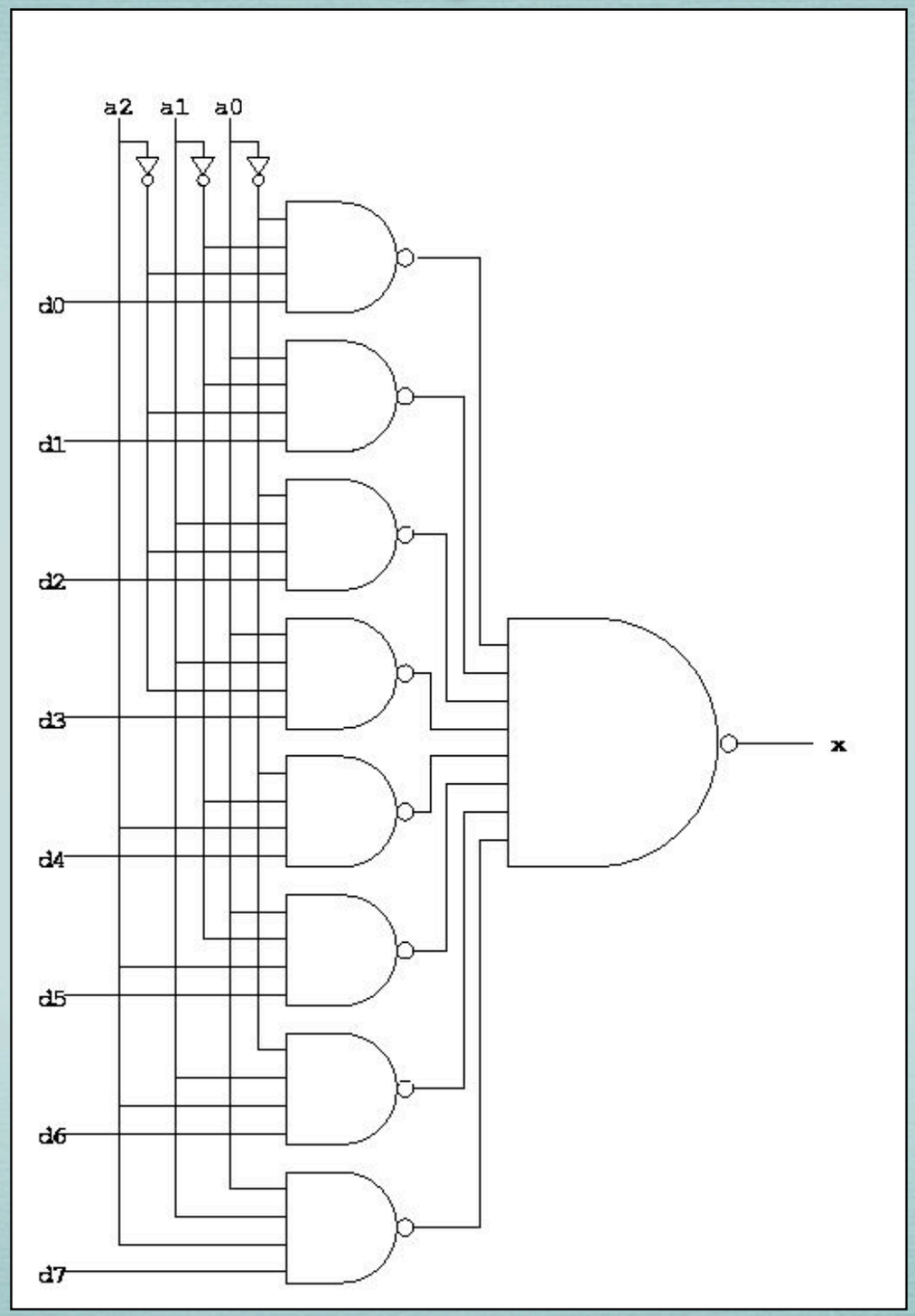






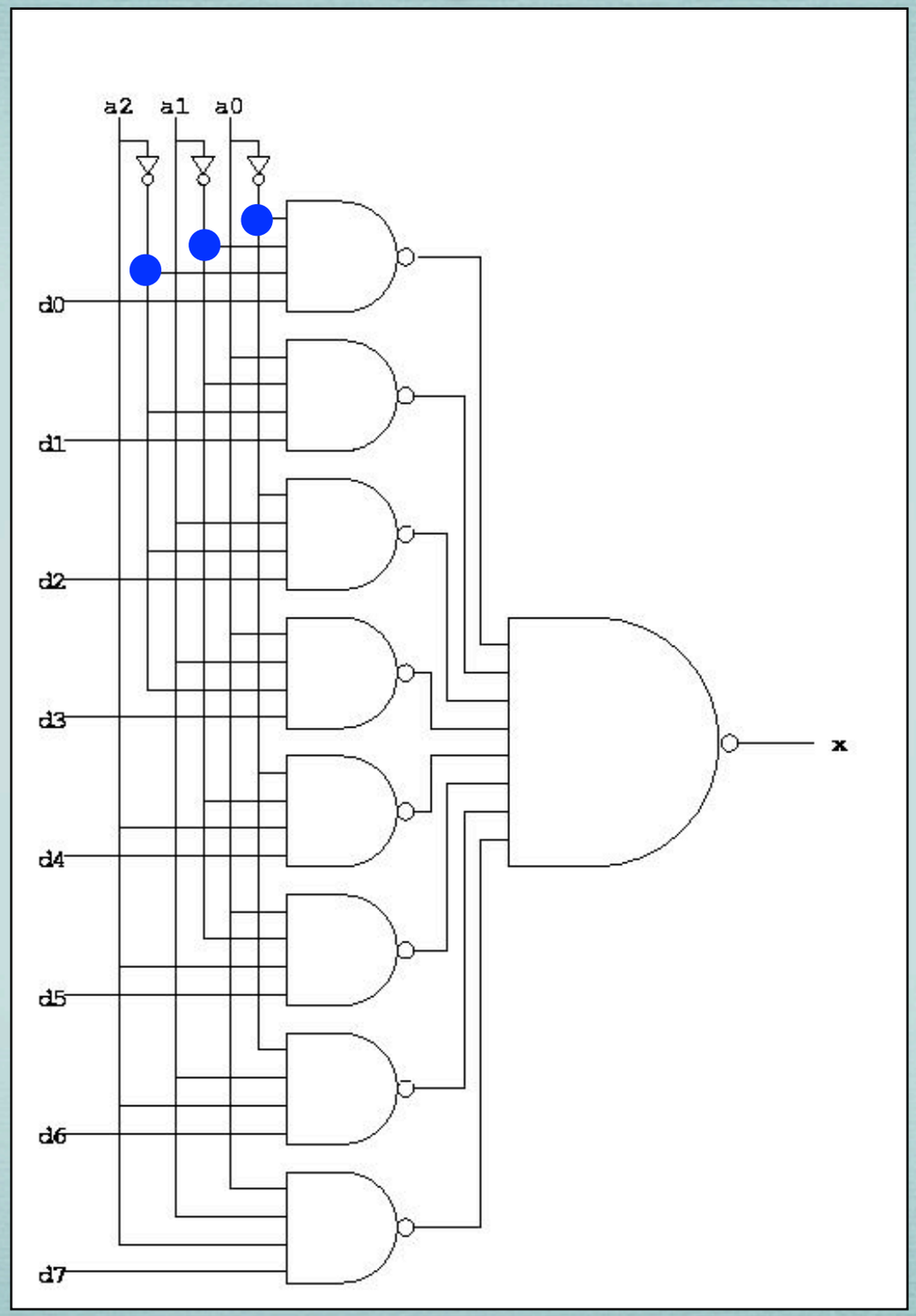


Multiplexer



Multiplexer

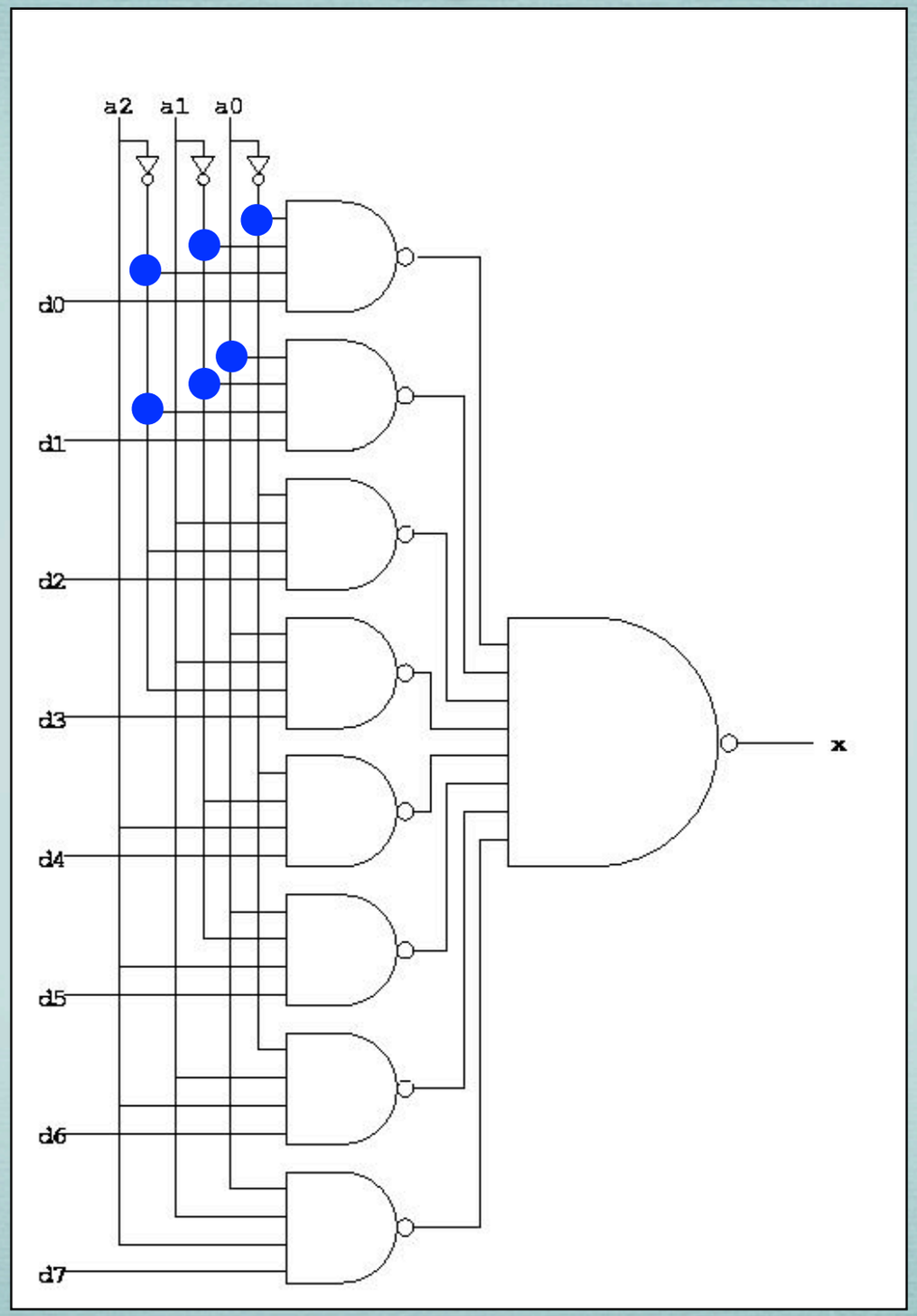
000



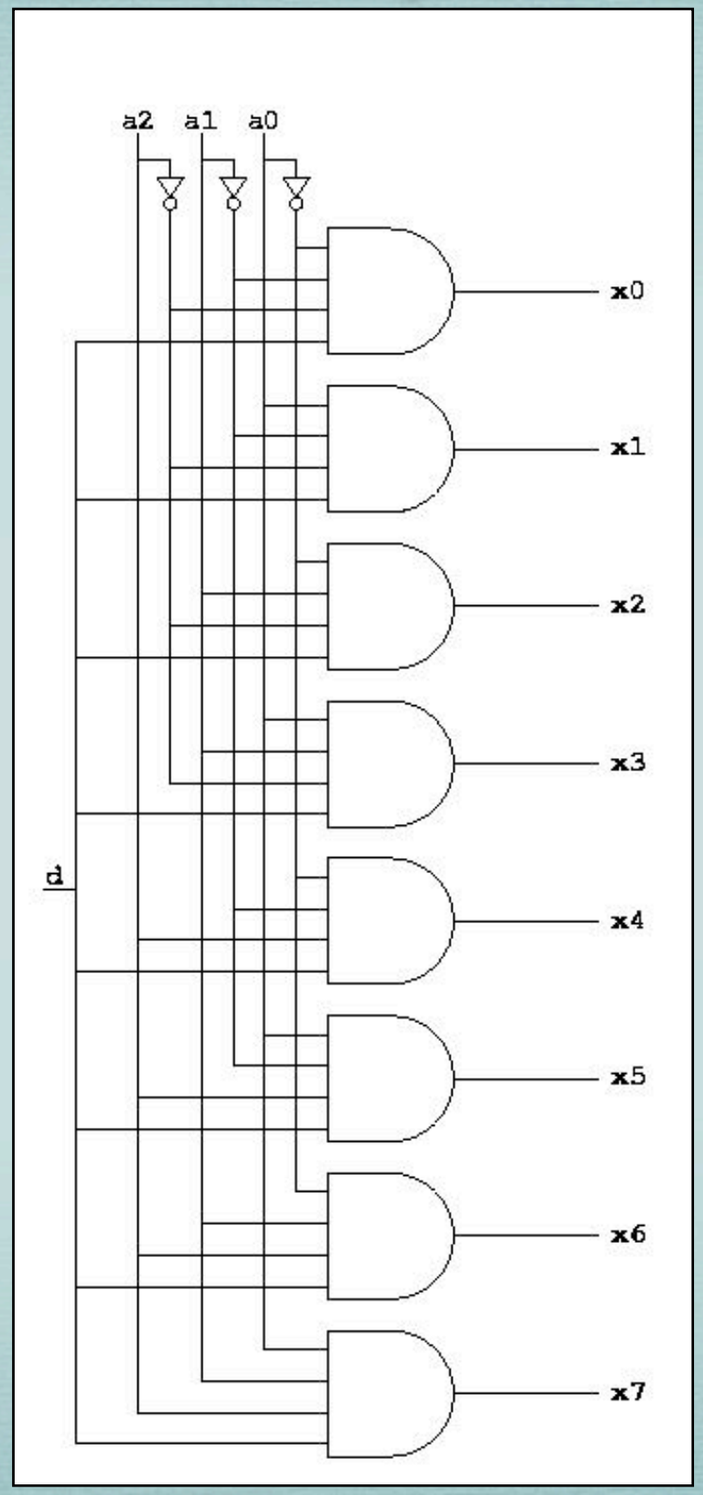
Multiplexer

000

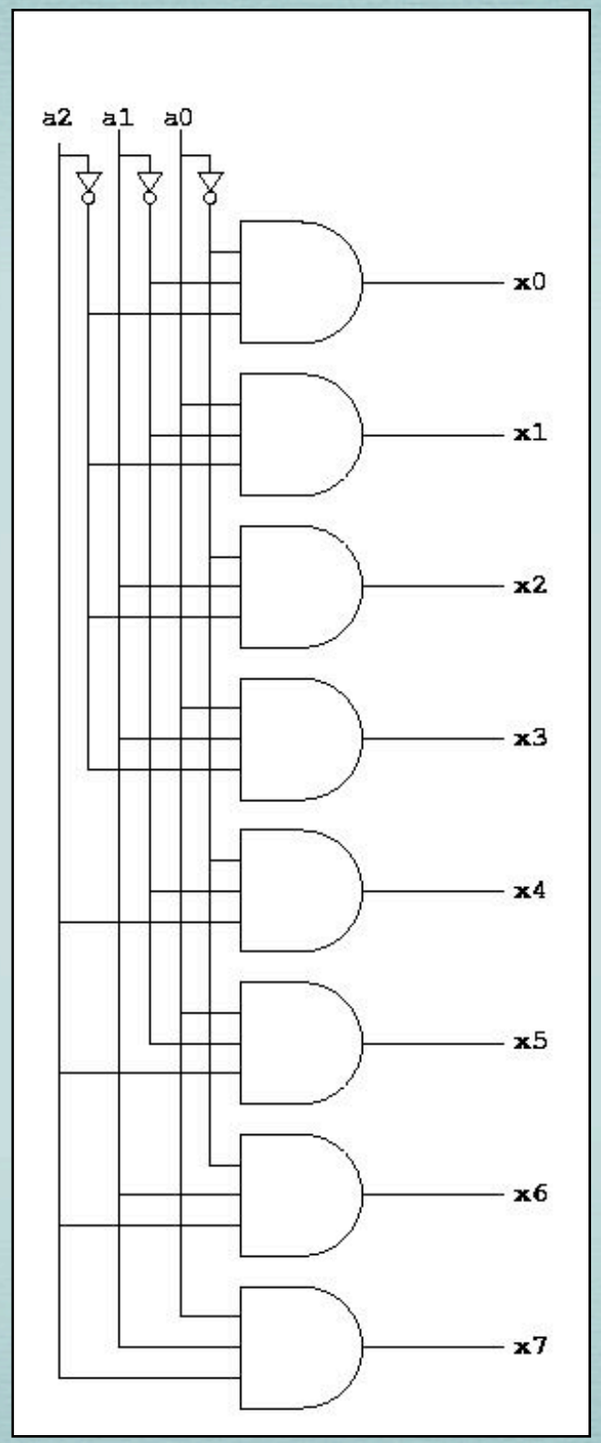
001



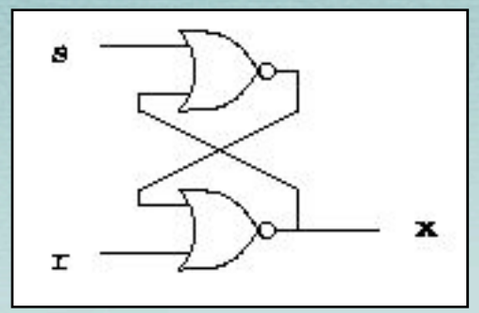
Demultiplexer

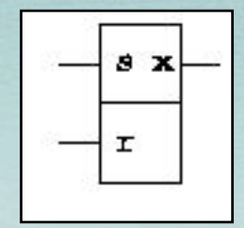


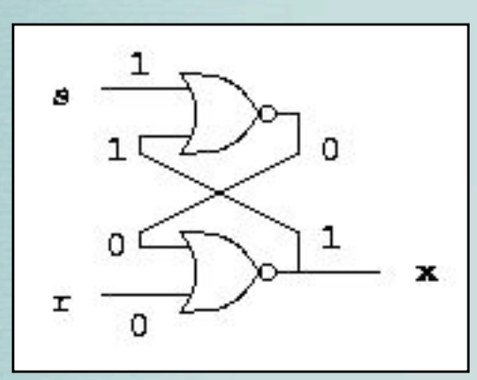
Decoder

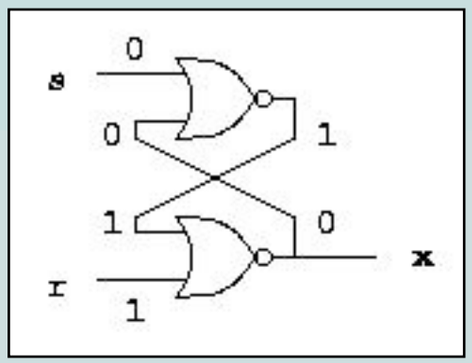


SR Latch



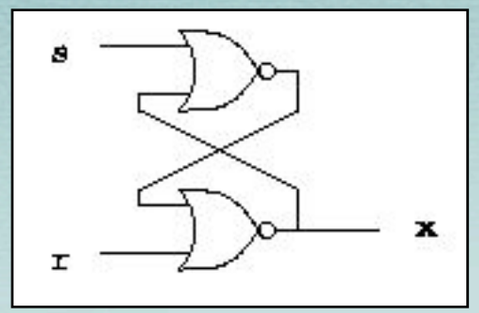


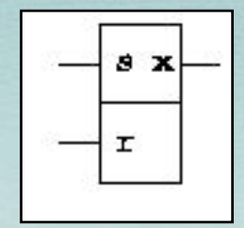


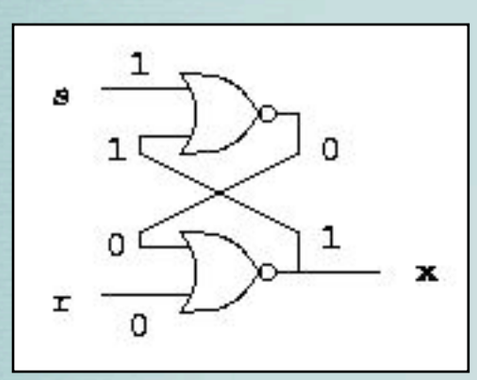


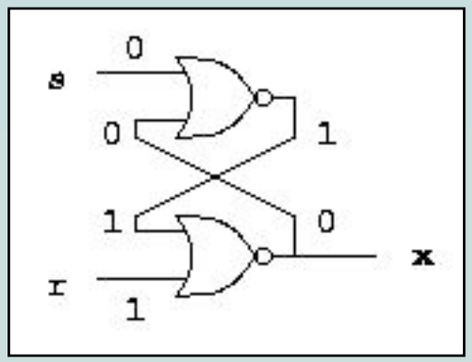
At most one input is 1

SR Latch

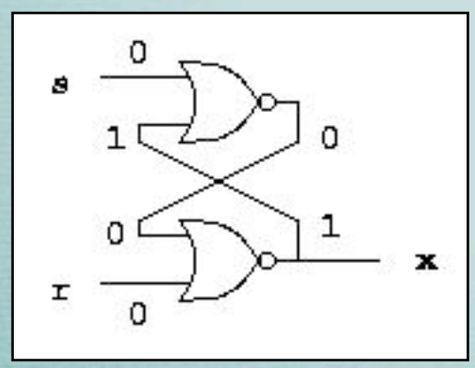


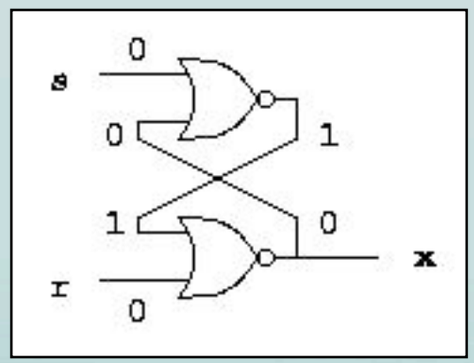






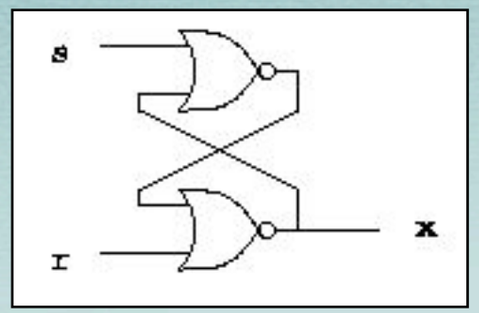
At most one input is 1

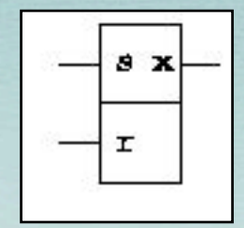


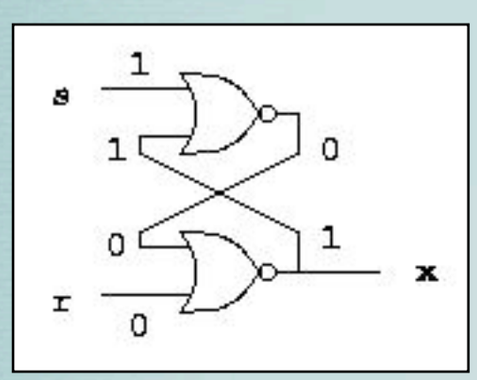


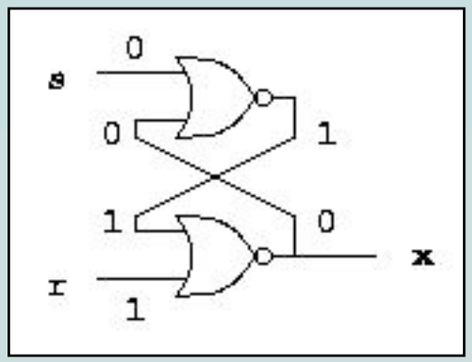
Latch “remembers”

SR Latch

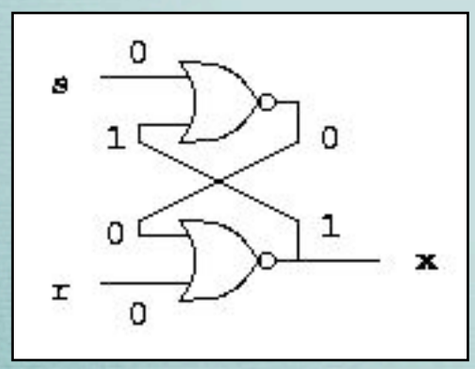


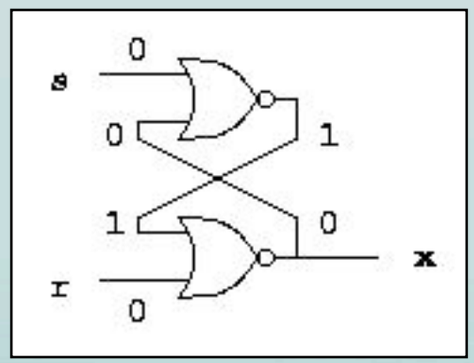






At most one input is 1

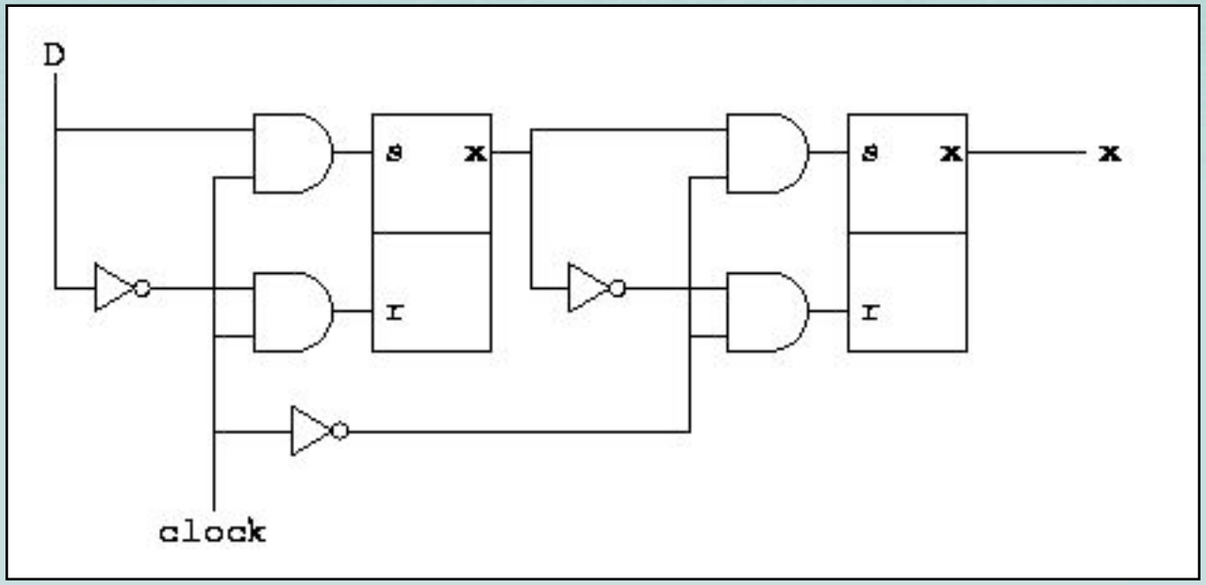


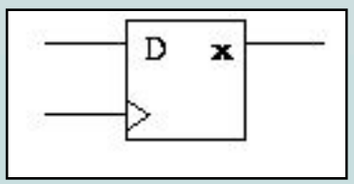


Latch “remembers”

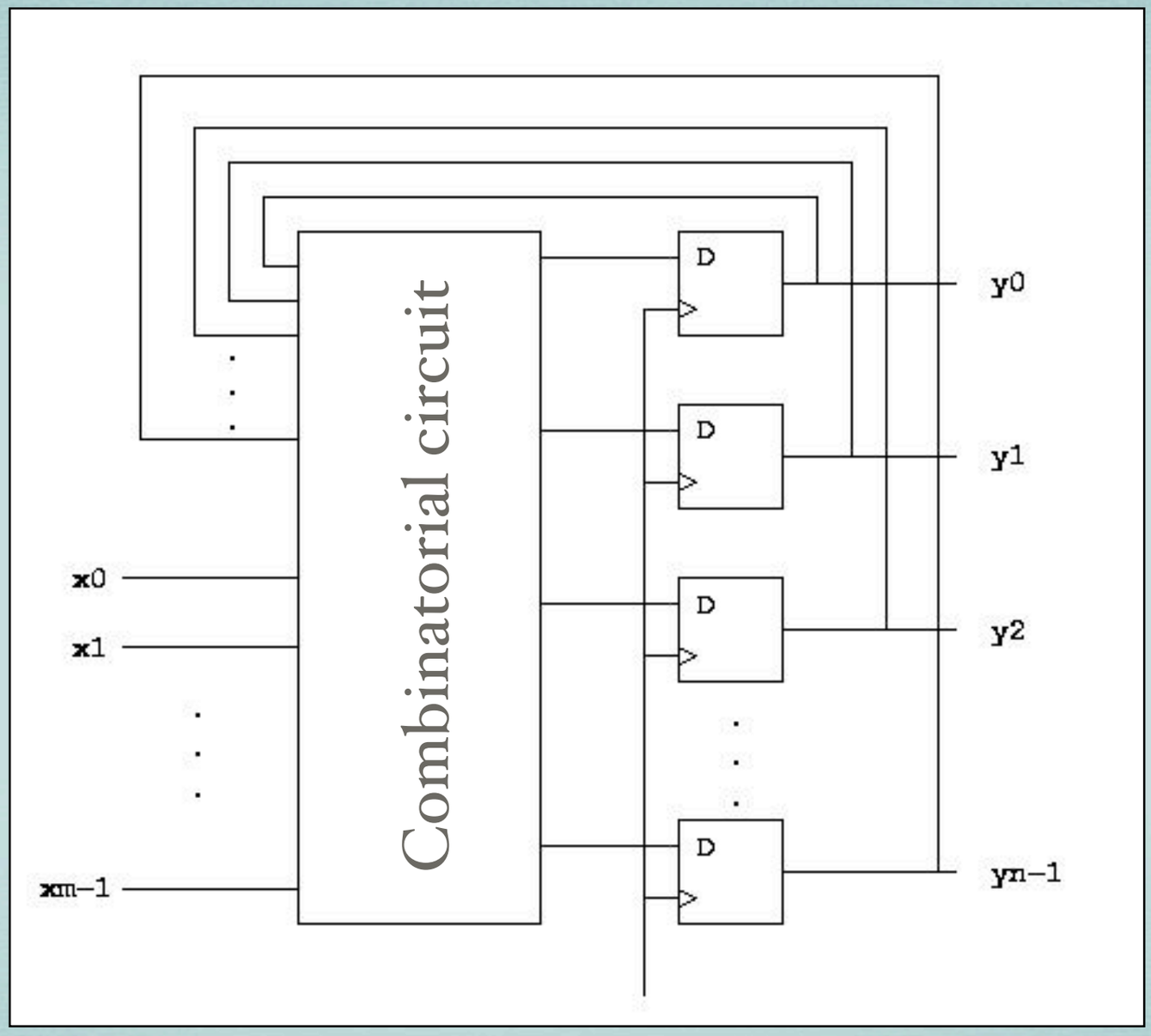
What if both the inputs are 1?

Flip-flops (Master-slave)

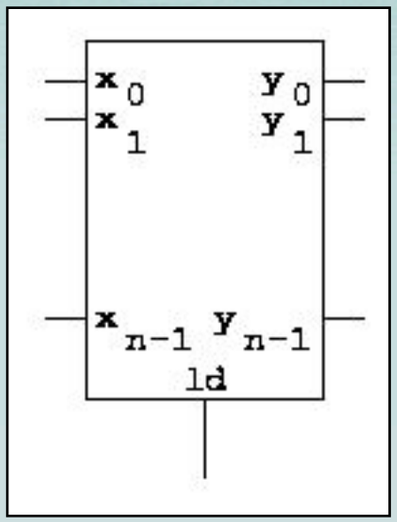




Sequential Circuits: General Design



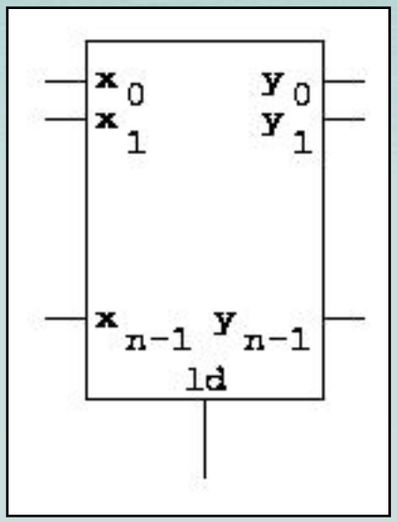
Registers



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

State table

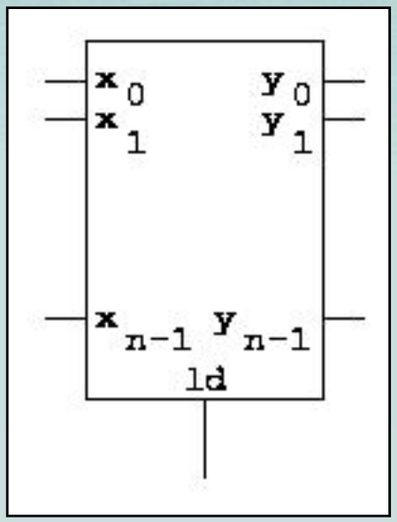
Registers



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

State table

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

State table

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