ECE 385 – Digital Systems Laboratory

Lecture 1 – Introduction and Lab 1

Zuofu Cheng

Spring 2019

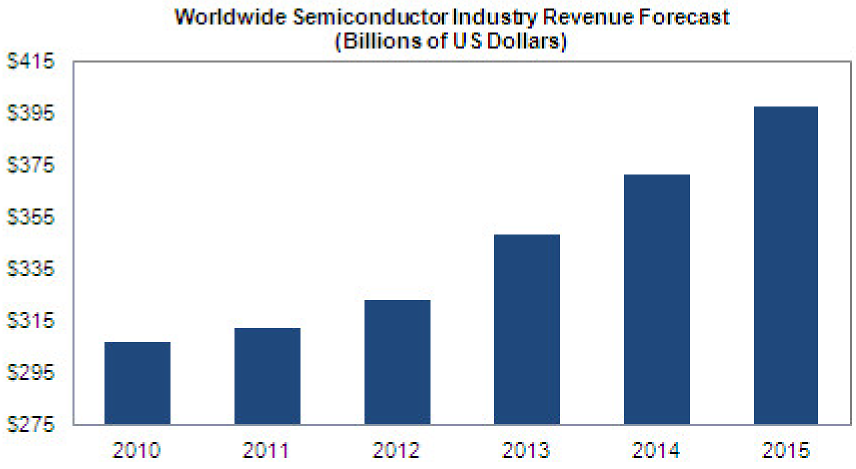
Link to Course Website

ECE ILLINOIS

ILLINOIS

The Booming of Semiconductor Industry

- **$350 billion industry today**
- **Electronics permeate almost every aspects of our lives**
  - Cell phones, cars, buildings …
  - Medical, social, e-commerce …
  - Military, security, aerospace …



Source: IHS iSuppli Research, January 2012

SAMSUNG

STMicroelectronics

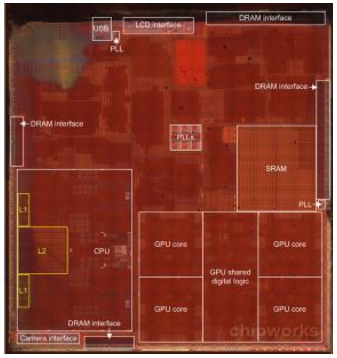
GLOBALFOUNDRIES

IBM

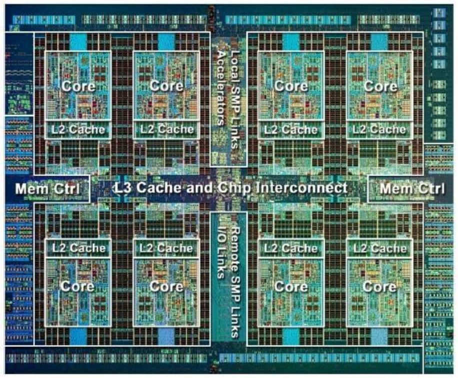
tsmc

TEXAS INSTRUMENTS

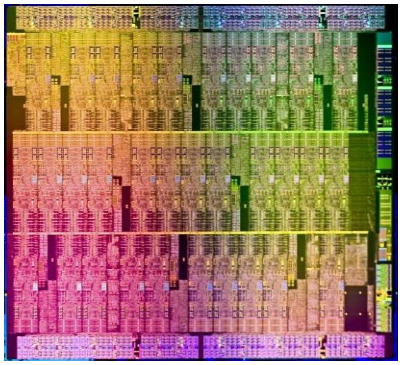
Billion-Transistor Chips



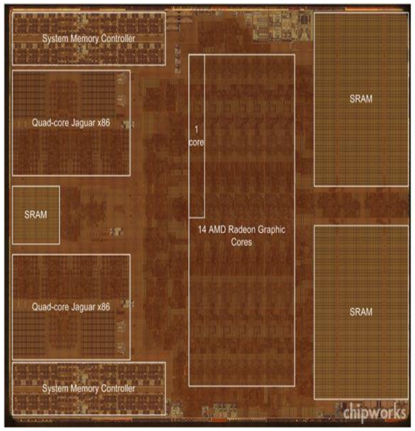
Apple A7
Over 1B transistors"



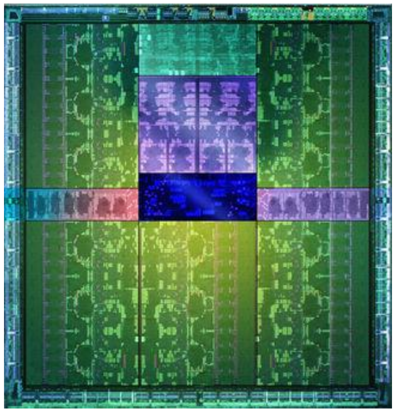
IBM Power7+
~2.1B transistors"



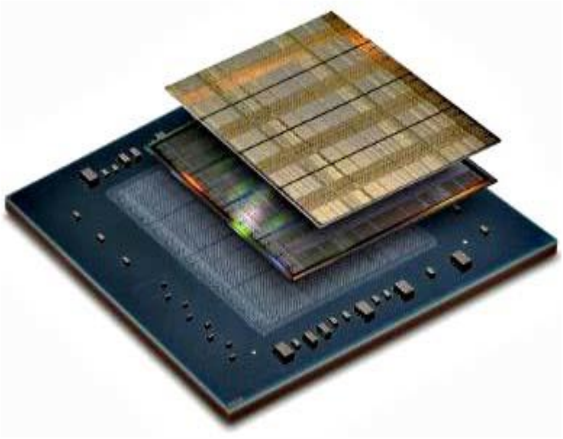
Intel Xeon Phi
~5B transistors"



Microsoft/AMD Xbox One
~5B transistors"

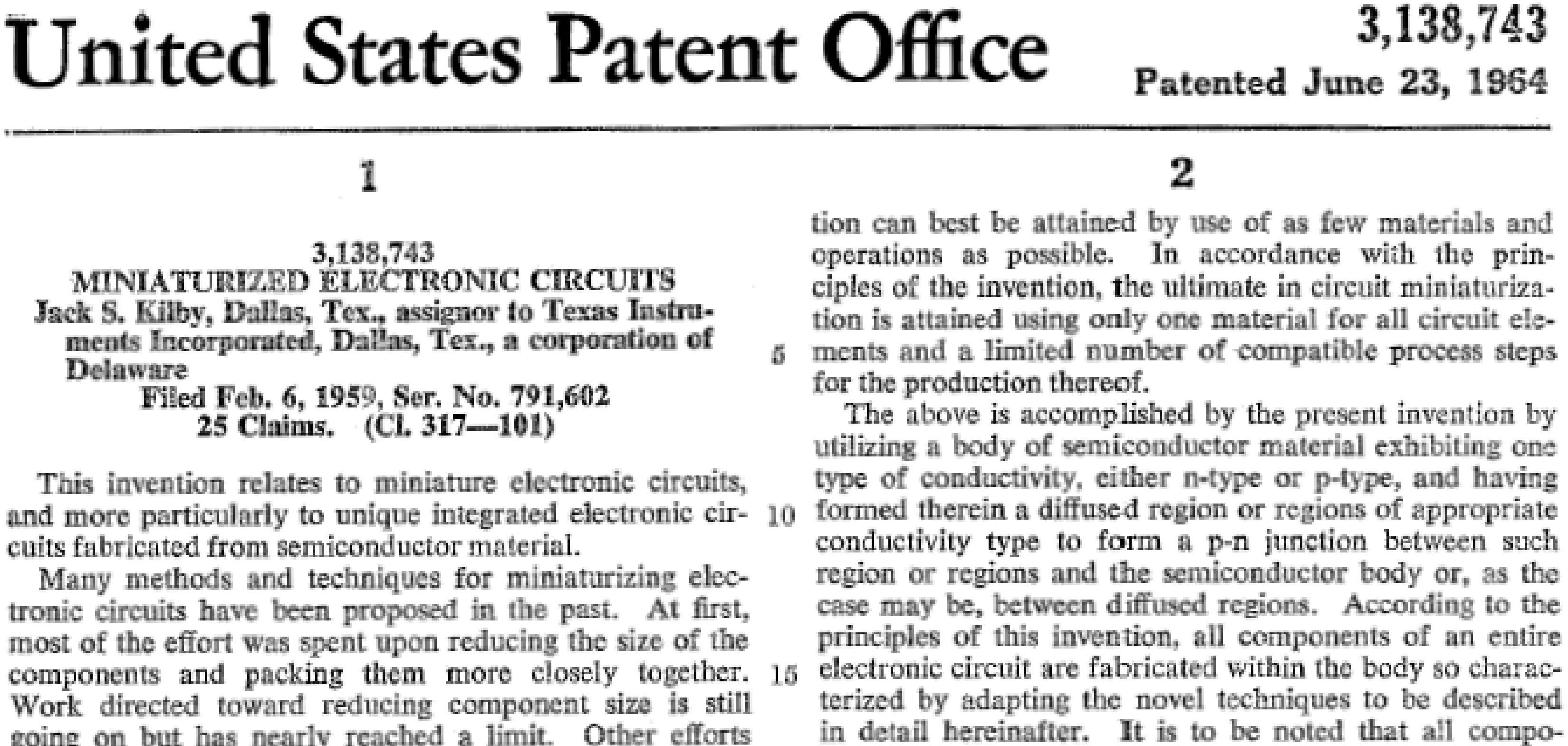


NVIDIA GK110
~7.1B transistors"



Xilinx Virtex-7
~6.8B transistors"

Source: Zhiru Zhang, Cornell



United States Patent Office

3,138,743

Patented June 23, 1964

1

3,138,743

MINIATURIZED ELECTRONIC CIRCUITS

Jack S. Kilby, Dallas, Tex., assignor to Texas Instruments Incorporated, Dallas, Tex., a corporation of Delaware

Filed Feb. 6, 1959, Ser. No. 791,602

25 Claims. (Cl. 317—101)

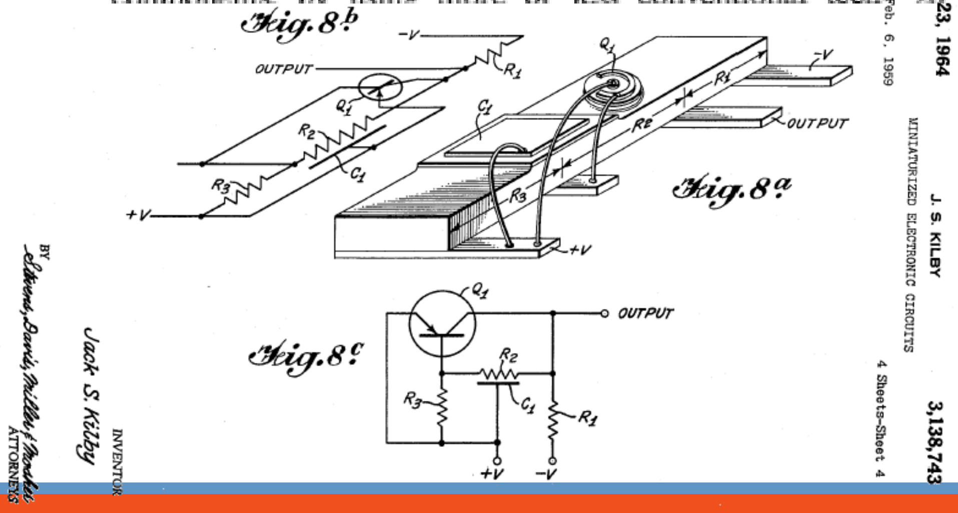
This invention relates to miniature electronic circuits, and more particularly to unique integrated electronic circuits fabricated from semiconductor material.

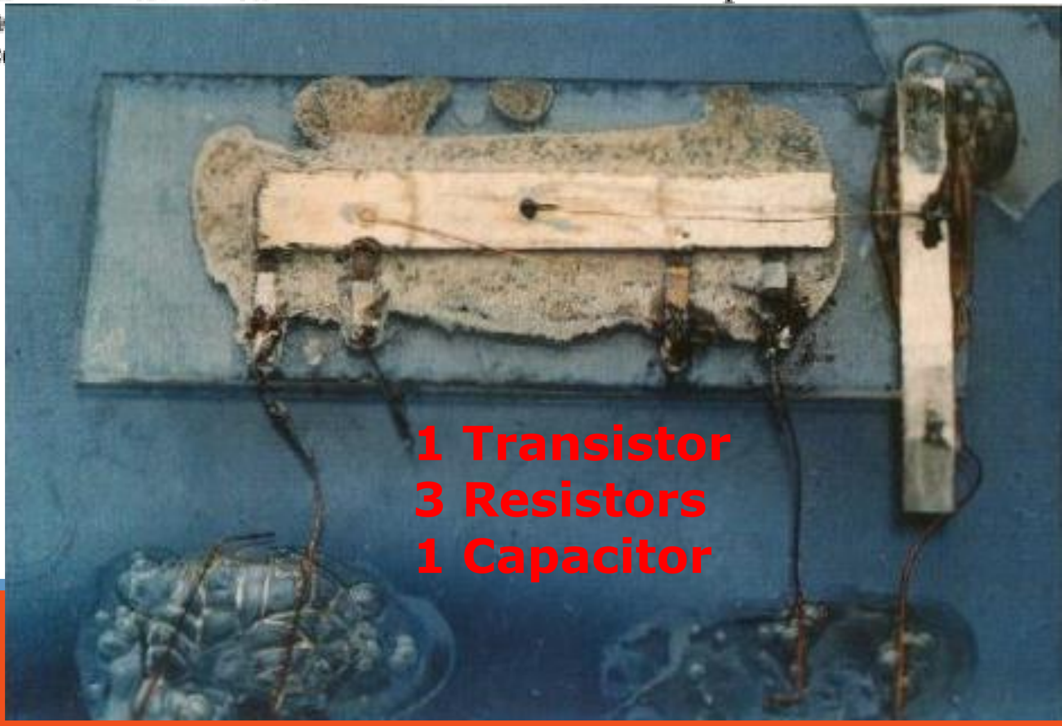
Many methods and techniques for miniaturizing electronic circuits have been proposed in the past. At first, most of the effort was spent upon reducing the size of the components and packing them more closely together. Work directed toward reducing component size is still going on but has nearly reached a limit. Other efforts have been made to reduce the size of electronic circuits such as by eliminating the protective coverings from components, by using more or less conventional tech-

2

tion can best be attained by use of as few materials and operations as possible. In accordance with the principles of the invention, the ultimate in circuit miniaturization is attained using only one material for all circuit elements and a limited number of compatible process steps for the production thereof.

The above is accomplished by the present invention by utilizing a body of semiconductor material exhibiting one type of conductivity, either n-type or p-type, and having formed therein a diffused region or regions of appropriate conductivity type to form a p-n junction between such region or regions and the semiconductor body or, as the case may be, between diffused regions. According to the principles of this invention, all components of an entire electronic circuit are fabricated within the body so characterized by adapting the novel techniques to be described in detail hereinafter. It is to be noted that all compo-

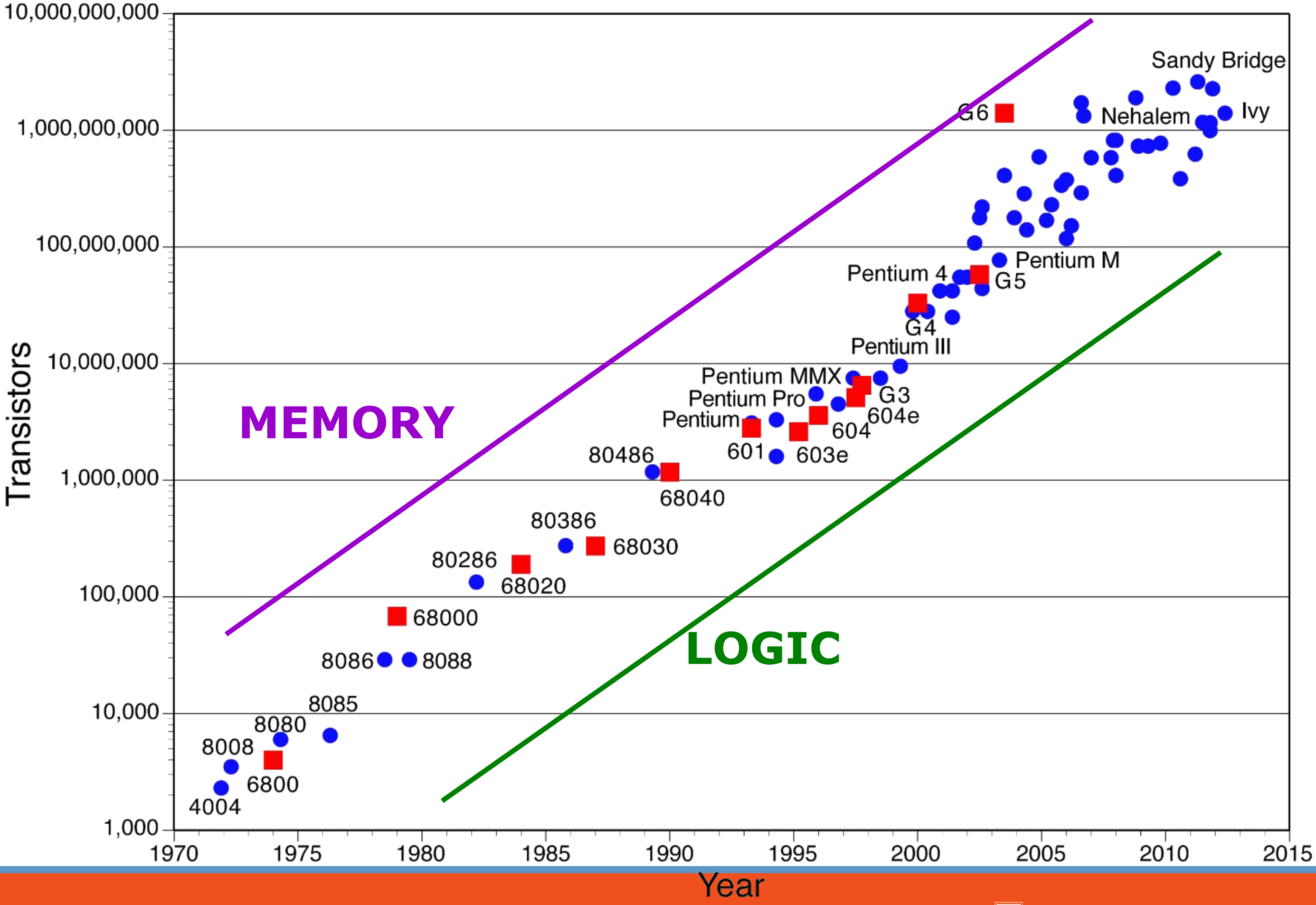




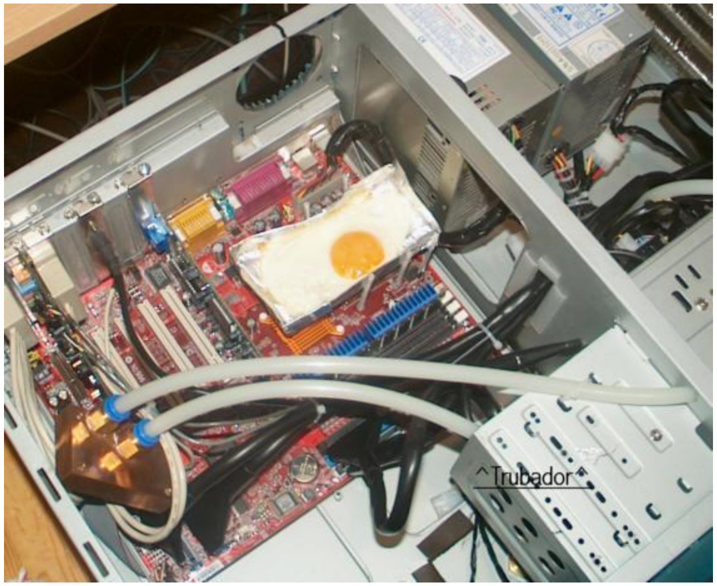
1 Transistor

3 Resistors

1 Capacitor


10,000,000,000
1,000,000,000
100,000,000
10,000,000
1,000,000
100,000
10,000
1,000
Transistors
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015
Year
MEMORY
LOGIC
4004
8008
6800
8080
8085
8086
8088
68000
80286
68020
80386
68030
80486
68040
Pentium
601
603e
Pentium Pro
604
Pentium MMX
604e
G3
Pentium III
G4
Pentium 4
G5
Pentium M
G6
Nehalem
Sandy Bridge
Ivy

Good Digital Design is Hard/Interesting!



FDIV Pentium bug cost 500 million dollars!

Power density of a processor higher than a stove!

Course Goal - Design, Construction, and Debugging of Digital Circuits

- Prerequisites from ECE 110: KVL, KCL, basic circuit analysis, resistors, transistors, LEDs, breadboard wiring and instrument usage
- Prerequisites from ECE 120: Boolean algebra, K-maps, combinational and sequential logic, storage elements, CPU building blocks (registers, ALU…)
- Prerequisites from ECE 220: C programming, CPU organization, basic data structures, I/O methods

Course Organization – Overview of Topics

- Hardware design using digital logic chips (TTL) and breadboard (labs 1-3)
- Design on the FPGA (Altera Cyclone IV) using SystemVerilog HDL (Hardware Description Language) (labs 4-6)
- Hardware and software co-design using FPGA and embedded CPU (labs 7-9)
- Final Project using FPGA board

Course Organization – Schedule

Week	Lab to Demo	Notes
1	None – Lab session starts Tuesday. form groups in lab	No demo for lab 1, individual lab report, work on lab 2 demo outside of class
2	2 (TTL storage)	Groups must be formed to demo lab 2, work on lab 3
3	3 (Bit-serial processor)	Install Quartus and do Quartus and SystemVerilog Tutorial, work on lab 4
4	4 (Intro-SV - adders)	Demo on FPGA board, work on lab 5
5	5 (SV multiplier)	Work on lab 6 week 1
6	6 (Week 1 SLC-3 CPU)	Demo lab 6 week 1, no extensions, lab 6.1 demo must be completed this week for credit, work on lab 6 week 2
7	6 (Week 2 SLC-3 CPU)	Demo lab 6 week 2, no demos for previous week's points, first midterm in class, work on lab 7
8	7 (SoC with NIOS II)	Work on lab 8
9	8 (Drivers on NIOS II for USB host and VGA)	Work on lab 9 week 1
10	No lab	Spring Break
11	9 (Week 1 AES software encryption)	Demo lab 9 week 1, no extensions, lab 9.1 demo must be completed this week for credit. Work on lab 9 week 2 and final proposal
12	9 (Week 2 AES hardware decryption)	Demo lab 9 week 2, no demos for previous week's points, final project starts (final proposal due)
13	Optional checkpoint	Second midterm in class, work on final project
14	Required checkpoint	Work on final project
15	Optional checkpoint	Work on final project
16	Final project demos	Return kits after demo, turn in final report within 1 week

Grading Breakdown - Labs

	report	demo	total
experiment 1	15	0	15
experiment 2	15	5	20
experiment 3	15	5	20
experiment 4	15	5	20
experiment 5	15	5	20
experiment 6	20	10	30
experiment 7	15	5	20
experiment 8	15	5	20
experiment 9	20	10	30
Total for 9 labs			195

- Reports due on Compass (electronic upload) at 11:59 PM day of next lab (**per group, except for lab 1 which is individual)**
- All codes must be handed into TA on USB flash drive at time of demo if any demo points are attempted

Grading Breakdown – Final Project, Exams

- For the 4 week-long Final Project (60 points)
  - 30 points for the proposal and written report
  - 20 points for demonstration
  - 10 points for difficulty
  - See Final Project page on Wiki for details

- Midterm I (40 minutes long – in class 2/25) (30 points)
  - Sample midterm problems are online (newer set is much more reflective of current exams)
- Midterm II (40 minutes long – in class 4/8) (30 points)
- Recommendation for exams – **print out lecture slides and take notes!**
- Peer evaluation (5 points) – given by **your partner** in end-of-semester review (upload to Compass) *For reference, over 90% of peer evaluations last semester were 5/5*

- Total 195+60+30+30+5 = 320 points

- Per-section curve to account for differences in report grading between TA. Expect this to be small, e.g. < 5 points (~1.5%)
  - **Don’t expect this to turn a C into a B, etc**
  - Overall class median was >80% pre-curve last semester!

Required Materials

- ECE 385 Laboratory Manual
  - Purchase from ECE copy room (version 18.2)
  - Material also available on course website
- Course Website:
  - https://courses.engr.Illinois.edu/ece385
  - Check course website often for announcements, Q&As, and additional lab materials!
- Lab Kit & FPGA Development Board (DE2-115) - One per team of two persons
  - Provided in the first lab session
  - You are responsible for making sure lab kit and FPGA board does not get lost/stolen.
- Recommended Reference Books
  - Patt, Yale N., Sanjay J. Patel, and J. Patel. Introduction to Computing Systems: From Bits and Gates to C and Beyond, 2004.
  - John F. Wakerly, Digital Design: Principles and Practices, Prentice Hall, New Jersey, (Third Edition), 2000.
  - **Stuart Sutherland, SystemVerilog for Design: A Guide to Using SystemVerilog for Hardware Design and Modeling, Springer, New York, 2006.**
  - James. O. Hamblen and M. J. Furman, Rapid Prototyping of Digital Systems – A Tutorial Approach, Kluwer, 2001.
  - Stuart Sutherland and Don Mills, Verilog and SystemVerilog Gotchas, Springer, 2007.
  - **Weste, Neil, David Harris, and Ayan Banerjee. “CMOS VLSI Design." A circuits and systems perspective 11 (2005): 739.**

Laboratory Policies– 4072 ECEB

- Regular lab sections (time in the lab for which you are registered)
  - 11 sections, 11 – 170 minute weekly meeting (9 labs + mid checkpoint + final demo).
  - Maximum 16 teams per section (32 students)
  - Each meeting is intended for debugging, measurements and demonstration
  - Design, wiring, or coding **must be done outside the regular lab** before coming to the scheduled meeting
  - **Do not expect** to complete lab assignment within the 170 minutes without significant preparation
- Lab Demo
  - Pre-lab (if there is one) write-up is **required** to begin lab meeting
  - Demo points documented on wiki for each lab
  - You are responsible for convincing TA that you deserve partial demo points (come up with required tests)
  - Lab report due before session one week after lab meeting
- Only come to your own lab session for demo
  - This is a fairness issue so everyone has the same amount of time to work on each assignment
- **In response to feedback: Q/A Panel** (optional "3rd lecture")
  - Time: Friday at 4:00 PM in 3013 ECEB – largely for benefit of early in week sections
  - Q/A format, instructor(s) will be present.
  - **Students must come with questions about upcoming lab (next week)**
  - Session ends when there are no more questions

Laboratory Documentation - Prelab and Report

- A Pre-Lab write-up consists of
  - Demonstrate that you attempted to understand and implement goals in lab
  - Answers to questions, if asked in the experiment (e.g. lab 7)
- A Post-Lab write-up consists of
  - Goals of the lab assignment
  - Documentation regarding the design process (elaboration of pre-lab)
  - Written description of the operation of the circuit
  - A block diagram with details of each block
  - A logic diagram if applicable
  - A component-layout diagram if applicable
  - Answer specific questions if asked in lab prompt
  - Measurements and analysis of final design (any waveforms, state diagrams, K-maps, simulation results, etc)
- Complete written lab report
  - Pre and Post Lab write-ups plus additional details
  - Due at the beginning of the next lab (upload to Compass before deadline)

Partnership, Late Policy, Academic Honesty

- Partnership
  - Partners share the same lab scores and thus the same responsibilities
  - Choose partner carefully (make sure your partner has similar goals, expectations, work ethic)
  - Contact TAs immediately when a conflict or a lack of communication occurs (defined as no contact within 48 hours)
  - Those without lab partners should check “Lab Partner Wanted” page under Coursework on Wiki.
- Cheating Policy
  - Simple definition of cheating – taking someone else’s work for credit
  - Every time caught cheating, the student will receive
    - A zero score for the entire lab or the test.
    - A whole letter grade reduction from the student's final grade.
    - Being reported to the college upon any cheating incident.
  - Students will be asked to un-wire the board completely after demo (labs 1-3)
  - For the SystemVerilog/C codes students submitted after demo, we will use automated similarity detection tool
  - For lab reports, Compass has built in plagiarism detection (SafeAssign)
- Late Policy
  - No late policy for lab reports – upload what you have at 11:59 PM on day it is due and start working on next lab
  - Exceptions will be made for excused illness, family emergency, etc

Getting Help with Assignments

- Attending lectures are the first way to get help
  - Lab assignments in manual and website tell you **what** you need to do in a concise way, but do not tell you **how** to do it
  - Lectures provide an overview and suggested approach to each experiment
  - Think of assignment as a design specifications, lectures as design meetings with engineering manager, convey recommendations for approach based on experience
  - Feel free to ask questions on lab even if they seem specific, other students are likely to be in same situation
- Office hours, open lab times, comments section of wiki
  - Will be posted on website
  - Use comments section for each lab as a message-board, TAs and course staff will try to answer questions
- When asking for help, be prepared to answer – “What have you already tried?”
  - Examples of good questions:
  - “I don’t understand how component Y builds on top of component X”
  - “Why do we do X when Y seems like it would be easier?”
  - “I added component X, and now the circuit stopped working, but it worked as expected prior to that”
  - “When I try perform this test, my output looks like X when it should be Y”
  - “I followed the advice given in the (Lectures/Message-board/Office Hours) but my results are different because of X”

Office Hours

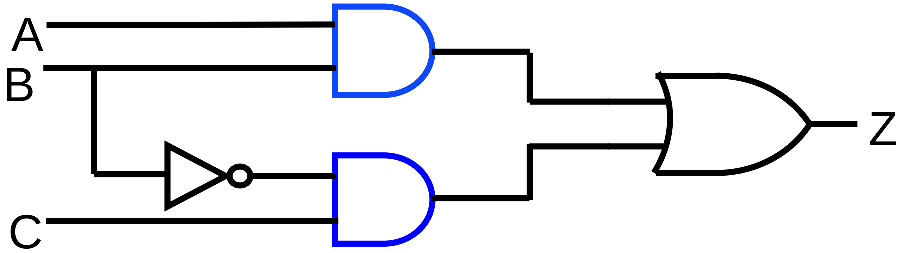
- Prof. Cheng's Office Hours
  - Monday, Wednesdays at 3:00 – 4:00 PM (4072/3022 ECEB)
- Open Labs (Undergrad TA Office Hours)
  - Check class schedule on wiki
  - In 4072 Lab for first three (TTL) labs, will move to 3022 for HDL labs
  - Plenty of coverage for all times there is no class in session (including nights and weekends)
  - This is in response to student feedback, there should be no excuse for not having labs done before demo
  - Open labs will get crowded! Make sure you build your own debugging hardware as in the General Guide

Experiment 1 – Introduction to TTL

- Goal – Design a 2 to 1 multiplexor using only quad 2-input NAND gate chips (7400)
- The naïve design will have glitches, redesign circuit to remove possibility of static-1 hazard
- Things to do in Lab 1 during session
  - Find lab partner in same section
  - Check out lab kit (one per group)
  - Build both naïve and glitch-free circuits
  - Capture operation using oscilloscope for comparison in lab report
- After your Lab 1 session
  - Write up lab report (this lab report is to be done **individually**)
  - Upload onto Compass before start of next lab session
- This will be the **only** time this semester will you should have enough time to complete the entire lab without preparation.

Experiment 1 – 2 to 1 MUX

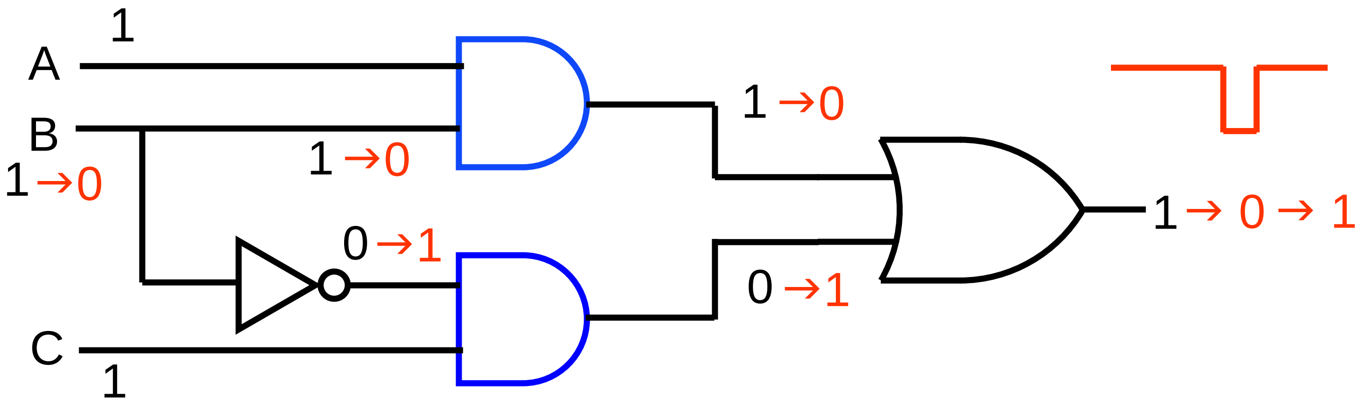
- What are inputs, outputs?
  - MUX can be thought of as digital switchbox, 2-1 means two inputs (A, C), one output (Z) and select (B)
- What is Boolean function for operation (in SOP form)?
  - Z = BA + B'C
- How do we implement using only NAND gates? How many 7400s do we need?

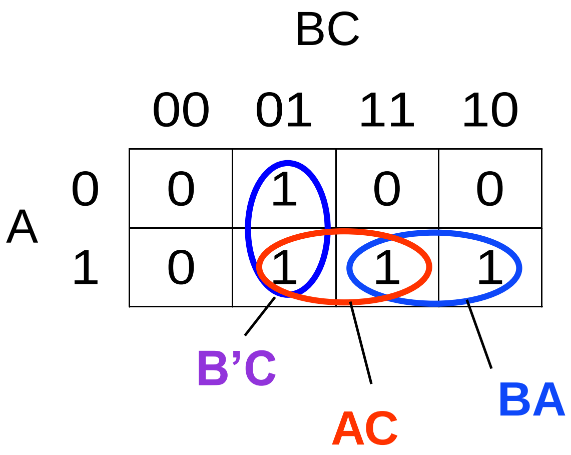


Static Hazards

- Boolean Algebraic analysis of a logic circuit gives us steady state (DC) behavior of that circuit. For example for a two-input OR gate, input of 10 and 01 both produce 1 output.
  - However, when the input 10 is changed to 01, both inputs may not change exactly at the same time in a physical world. There exists a possibility of both inputs assuming 0 values for a short time. This could lead to a short duration 0 output. This is called a glitch.
- A Static-1 Hazard is the possibility of producing a 0 glitch (pulse) when the static analysis of the logic tells us that the output should stay at logic 1.
  - Formal Definition: A Static-1 Hazard is a pair of input combinations that (a) Differ in only one input variable and (b) both give a 1 output; such that it is possible for momentary 0 output to occur during transition in the differing input variable.
- A Static-0 Hazard is the possibility of producing a 1 glitch (pulse) when the static analysis of the logic tells us that the output should stay at logic 0.
  - Formal Definition: A Static-0 Hazard is a pair of input combinations that (a) Differ in only one input variable and (b) both give a 0 output; such that it is possible for momentary 1 output to occur during transition in the differing input variable.

Static Hazards – Example with 2 to 1 MUX





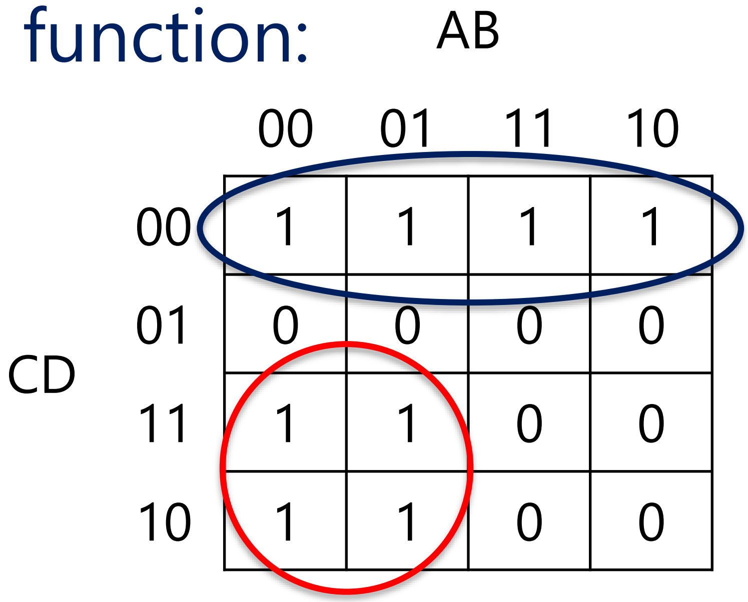
A \ BC	00	01	11	10
0	0	1	0	0
1	0	1	1	1

To avoid Static-1 hazard in an AND-OR (SOP) circuit, cover all adjacent min-terms in the K-map.
In this example, add the term AC

- What is the final Boolean function for glitch-free circuit?
  - Z = B'C + BA + AC

Static Hazards – Another Example

- Another example, consider 4 input function:
- What is minimal representation?
- Z = C'D'+A'C



AB

CD	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	1	1	0	0
10	1	1	0	0

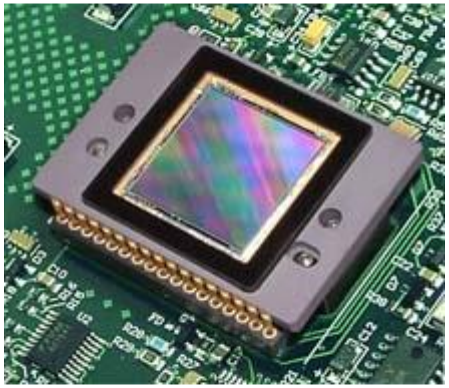
- Are there any static hazards?
  - Adjacent product terms without overlapping terms?
  - Remember, K-maps wrap around
- How do we solve static hazard? (what term(s) should we add?)

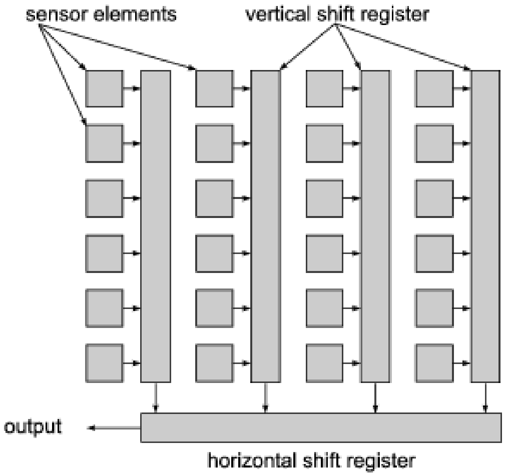
Experiment 2 – Data Storage with TTL

- Design a Shift Register Memory (4 x 2 bit)
  - Serial Input and Output
  - Circulating data
  - Use a shift-register chip (e.g. 74LS194A) and other logic
  - Data sheet of the chip is posted online (under Labs)
  - **Must not use parallel load or parallel output (only serial input and right-most output may be used)**
  - **Data must also be continuously shifting**

Experiment 2 – Motivation

- Why use shift register to create RAM?
  - Shift registers are simpler to layout and wire (no address pins, fewer data pins)
  - Some devices are inherently organized as shift registers (example CCD image sensor)

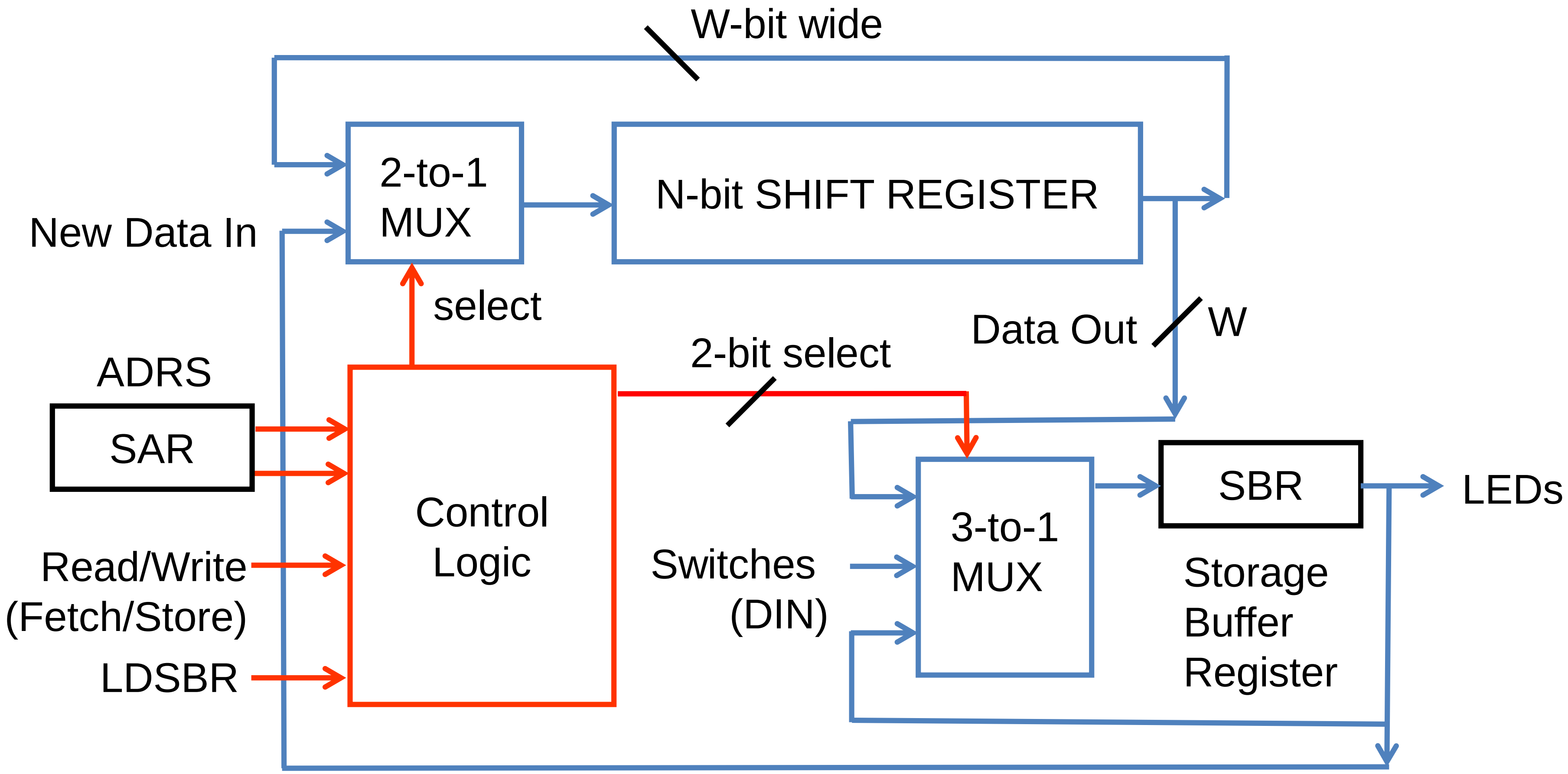




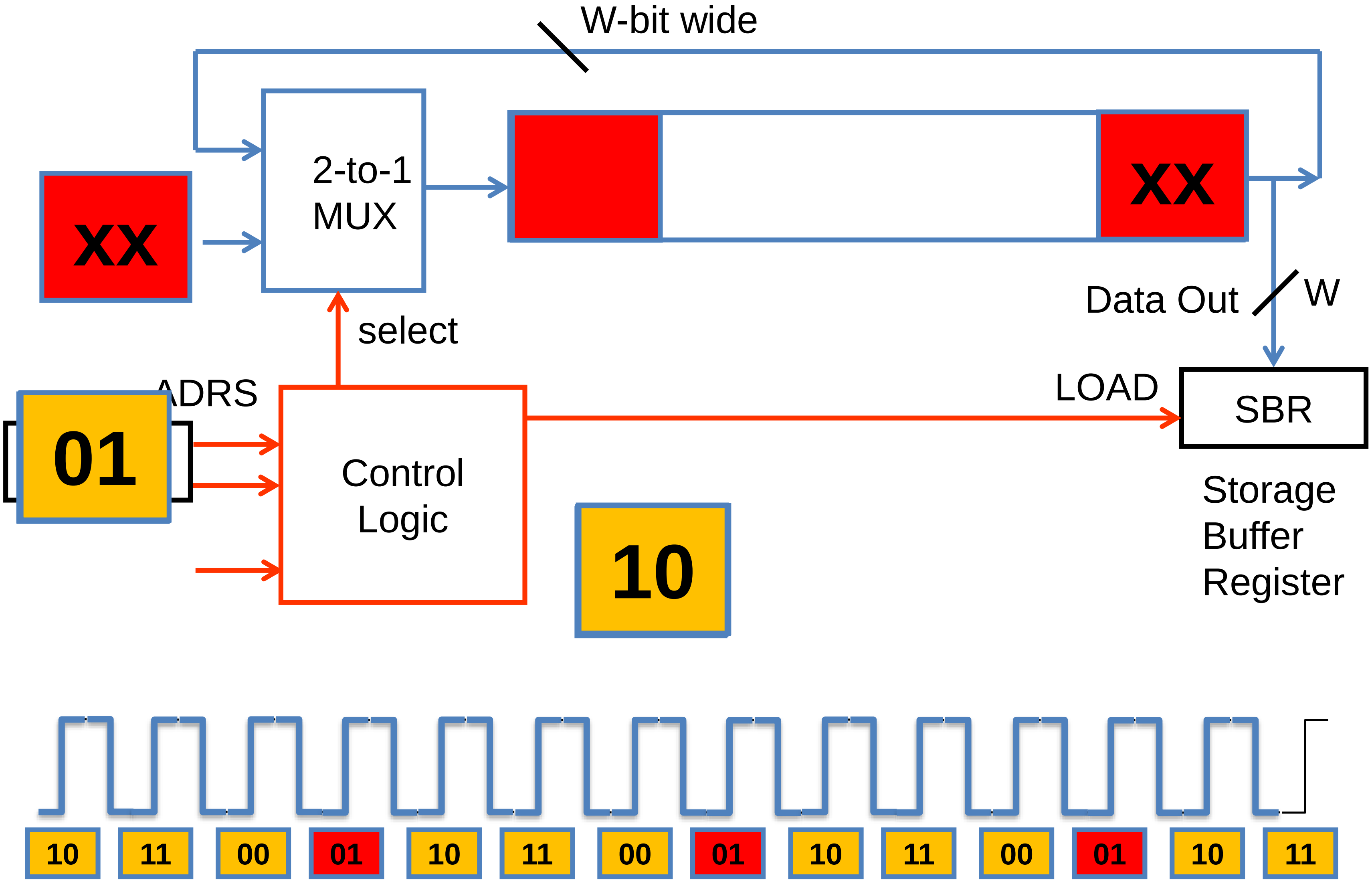
  - Note that any type of memory (shift register, parallel RAM, FIFO) may be used to emulate any other type, given additional control logic

Experiment 2 – Data Storage with TTL

Use two 74LS194 shift-registers
Serial Input and Output
Circulating data



Experiment 2 – Operation of Circuit



Experiment 2 – Inputs and Outputs

- FETCH (Switch)
  - When FETCH is high, the value in the data word specified by the SAR is read into the SBR.
- STORE (Switch)
  - When STORE is high, the value in the SBR is stored into the word specified by the SAR.
- SBR1, SBR0 (Flip-flops)
  - The data word in the SBR; either the most recently fetched data word or a data word loaded from switches
- SAR1, SAR0 (Switches)
  - The address, in the SAR, of a word in the storage
- DIN1, DIN0 (Switches)
  - Data word to be loaded into SBR for storing into storage
- LDSBR (Switch)
  - When LDSBR is high, the SBR is loaded with the data word DIN1, DIN0

Experiment 2 – Demo

- FETCH (3 points)
  - M[SAR] -> SBR (multi-cycle operation)
- STORE (1 point)
  - SBR -> M[SAR] (multi-cycle operation)
- LOAD (1 point)
  - DIN -> SBR (single-cycle operation)
- NOP
  - Do nothing, contents of memory is preserved (shift register must continue to shift – **do not use clock inhibit on shift register)**
- You can assume that only one of the FETCH/STORE/LDSBR switches will be set at any given time (behavior is undefined if more than one is set at one time)
- You can get 1 demo point for demonstrating proper shift operation (not necessary to independently demo if FETCH is completely working)

Experiment 2 – Hints

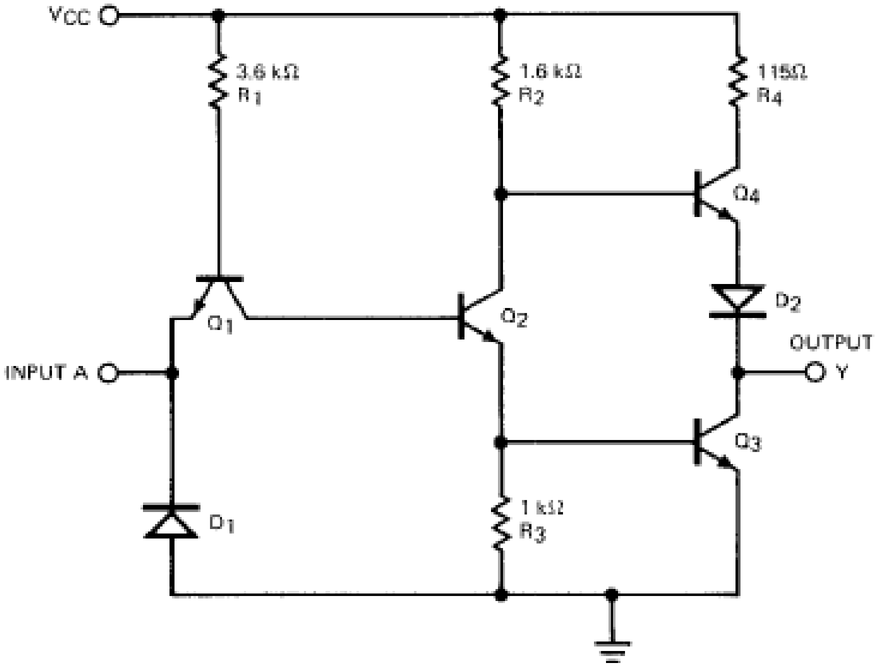
- Do
  - Use control logic to control when to write / fetch
  - Shift registers need to be continuously shifting
  - Inputs go to control logic, which then sends out signals to other parts of the circuit
  - To distinguish each storage location, each location needs to be assigned to a unique address
  - To identify the correct location to write / fetch, the address of the locations need to be compared against the desired address (SAR)
- Don’t
  - Don’t put logic on clock (do not gate clock – poor practice in synchronous designs – we will talk more about this later)
  - A single inverter to flip clock edge is ok

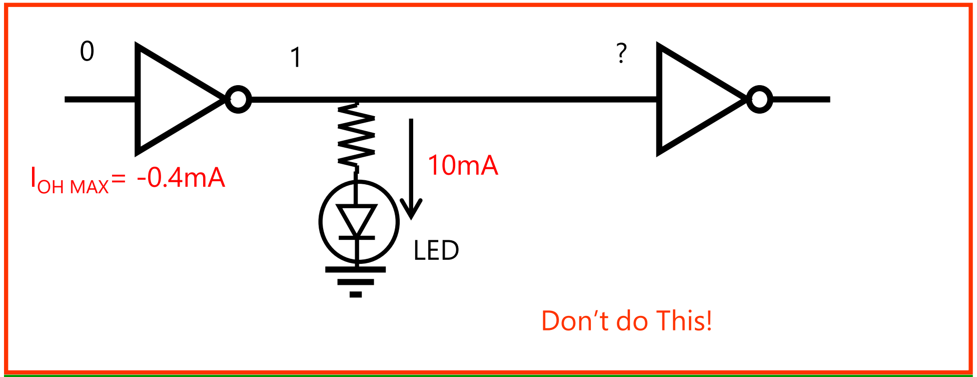
Experiment 2 – Debugging at Home

- You will be expected to come into lab with mostly working circuit
  - Oscilloscope / function generator is available in open-lab sessions
  - Lab kits include LEDs and switches for seeing outputs and inputs
  - LEDs must be driven with current limiting (with current sink driver)
    - Use inverter as driver
    - Use current sink topology to avoid inversion of LED
  - Switches must be de-bounced (or glitches will occur)
    - Only necessary for clock signal (if design is strictly synchronous)

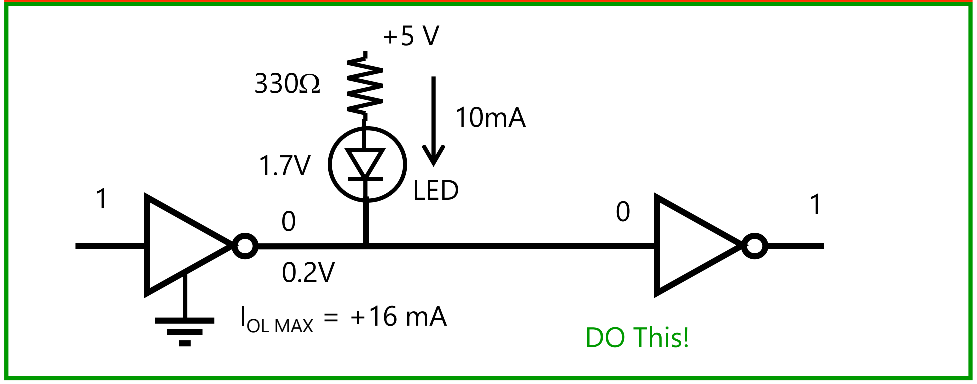
Experiment 2 – Driving LEDs with TTL

- TTL ICs have very limited current source capability (why? take a look at TTL inverter schematic below)
- Resistor and gate to pull **down** (note LED is on when output is





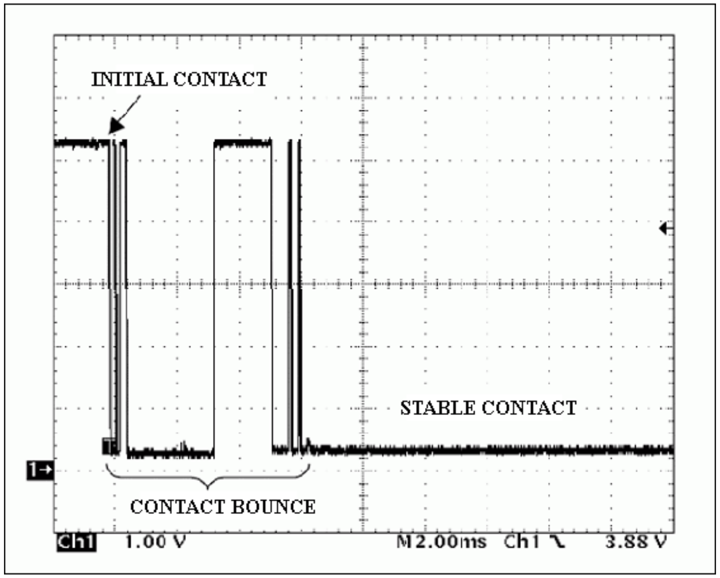
Don't do This!

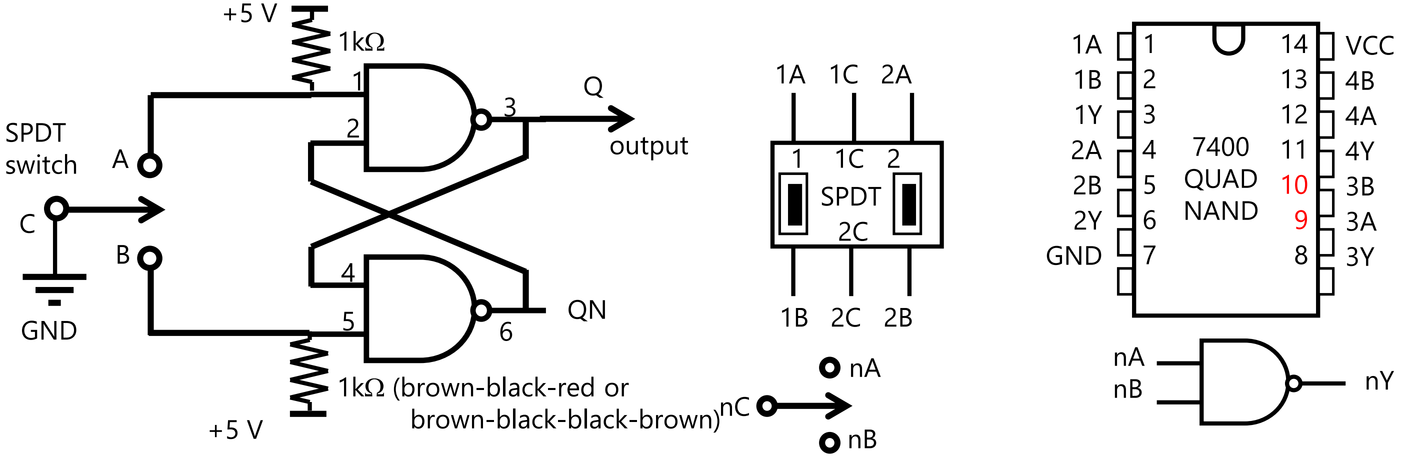


DO This!

Experiment 2 – De-bouncing Switches

- Mechanical switches will cause glitches during switch due to contact bounce


INITIAL CONTACT
STABLE CONTACT
CONTACT BOUNCE
1→
Ch1 1.00 V
M 2.00ms Ch1 ⌝ 3.88 V


+5 V
1kΩ
1
2
3
Q
output
SPDT
switch
A
C
B
GND
4
5
6
QN
1kΩ (brown-black-red or
brown-black-black-brown)
+5 V
1A 1C 2A
1 1C 2
SPDT
2C
1B 2C 2B
nA
nC
nB
1A 1 14 VCC
1B 2 13 4B
1Y 3 12 4A
2A 4 11 4Y
2B 5 10 3B
2Y 6 9 3A
GND 7 8 3Y
7400
QUAD
NAND
nA
nB
nY