A few midterm solutions…

For the following three questions you have an image processor that processes images line by line with the following steps:

Read line of image (1200 clock cycles) Process line (2000) Write new line of image (1200)

8. (2 points) If an image had 640 lines, what would be the throughput in images/clock cycle?

(1200 + 2000 + 1200) clks/line * 640 lines/image = 2816000 clocks/image

≈**3.55 E-7 images/clock cycle**

9. (2 points) If four processors could run in parallel on the same image, how would that affect the throughput? __________

a. It would have 0.5X the throughput b. It would have 2X the throughput

c. The throughput would be the same d. It would be 4x the throughput

10. (2 points) Imagine this processor was used for video, and there were two systems. System one had four processors working in parallel on the same image. System two had four processors, but each one worked on a different image simultaneously, which of the following statements would be true?

a. The throughput (in images/clock cycle) of system one would be faster

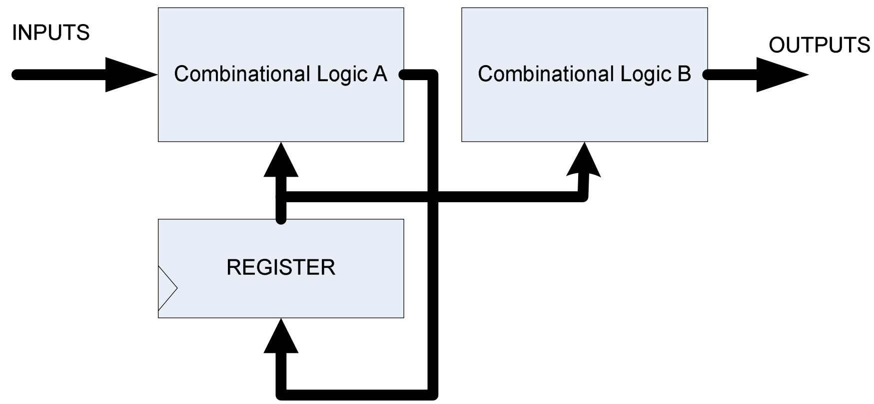
b. The throughput of system two would be faster.

c. The throughput would be the same for each, but the latency for one image to be processed would be faster for system one

d. The throughput would be the same for each, but the latency for one image to be processed would be faster for system two

A few midterm solutions…

13. (2 points) Here is the diagram for a simple Moore State Machine:



What is the purpose or 'Combinational Logic A'?
Determine the next state depending on the current state and the inputs.

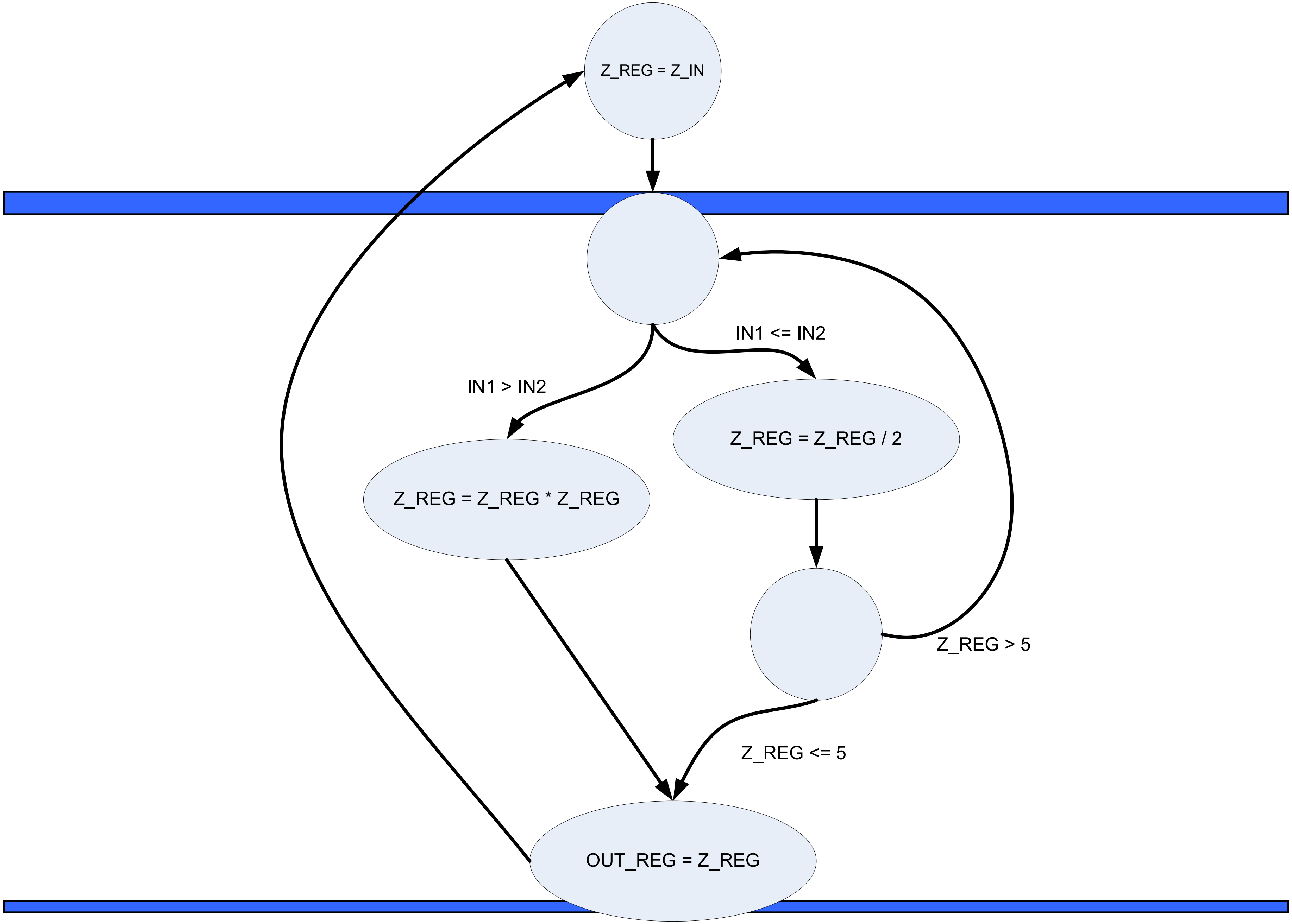
A few midterm solutions…

14. (2 points) In a general purpose processor, the address of the instruction to be read is held where?

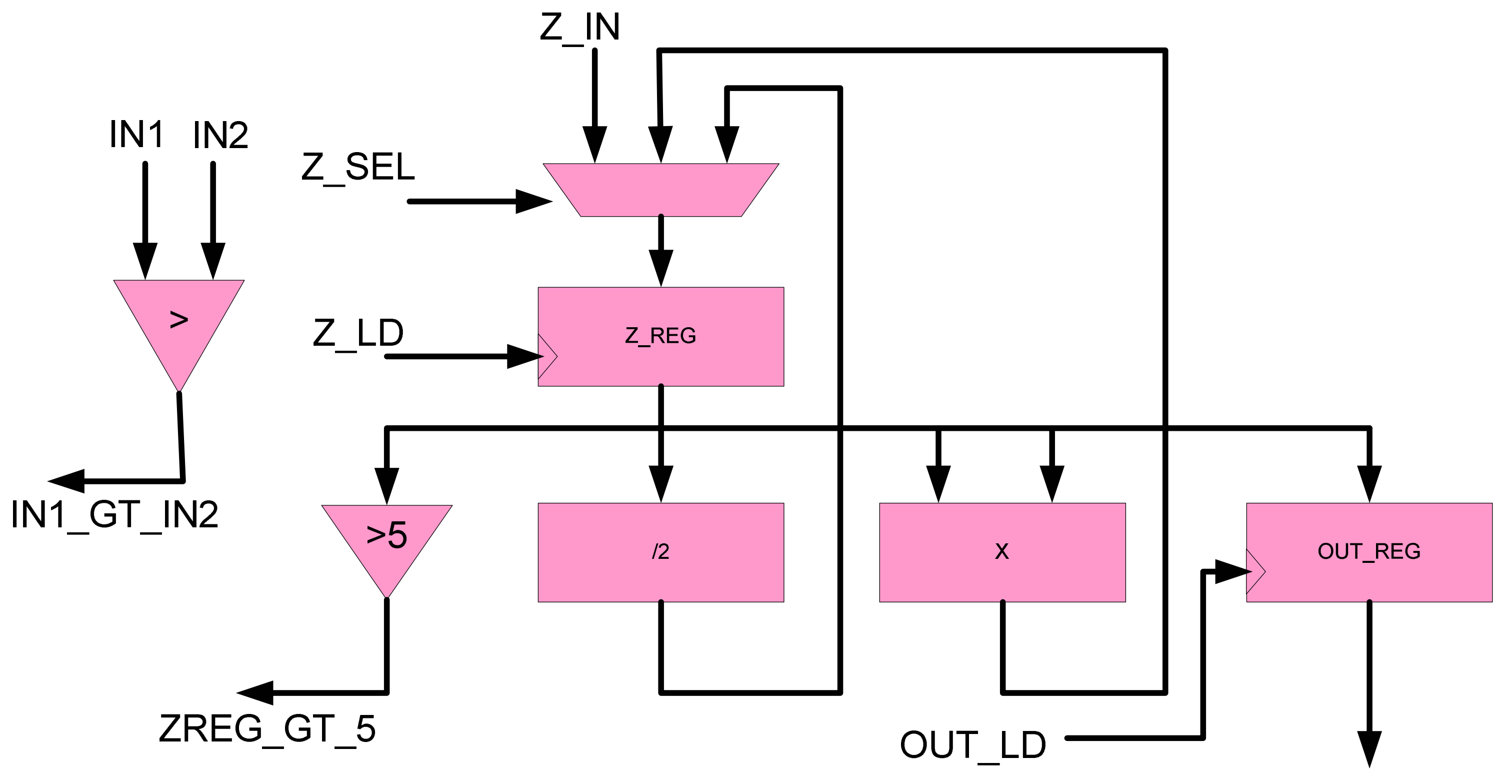
a) Program Counter b) Instruction Register c) ALU d) Accumulator

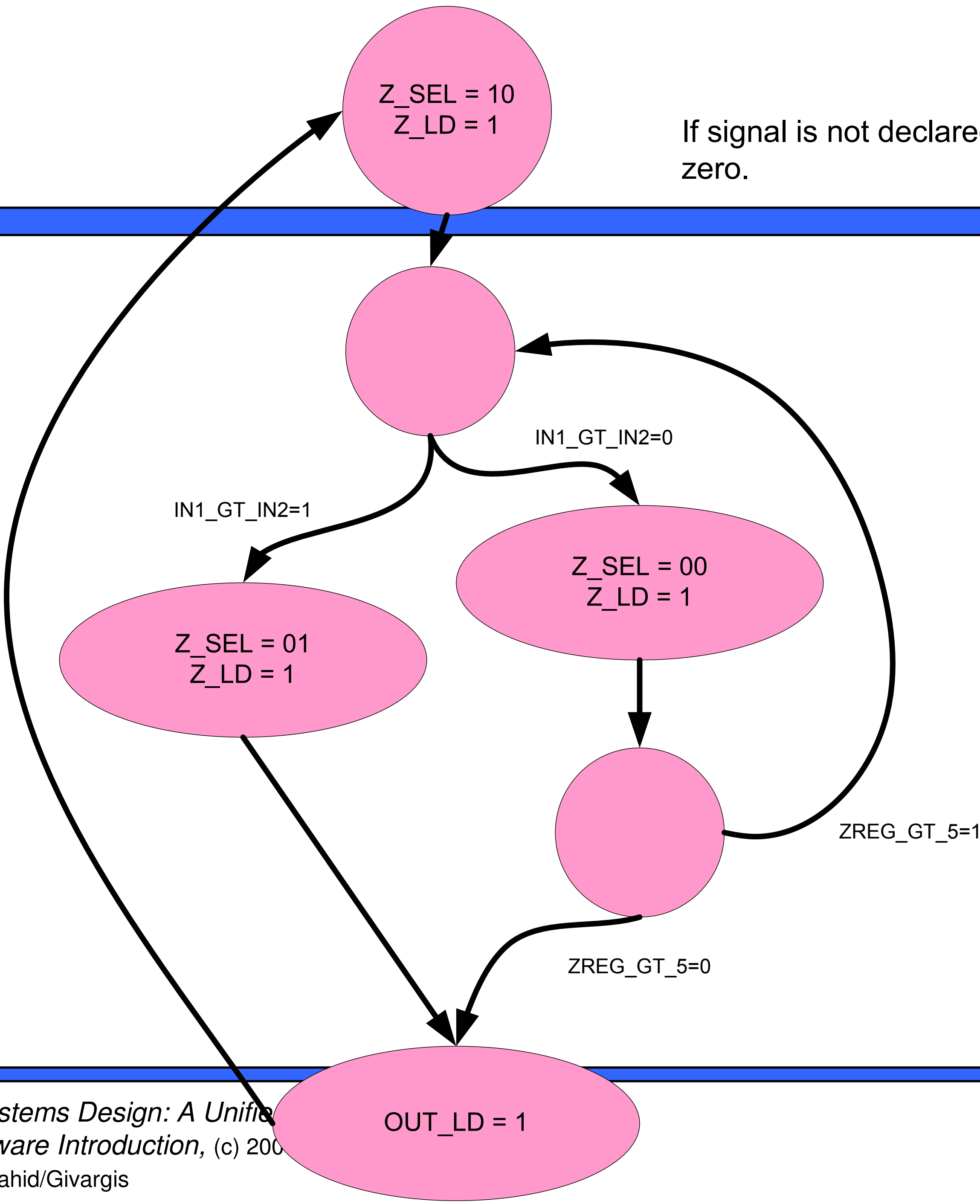
17. (2 points) If you cascade two counters so that the 'top' (the signal that indicates it has reached it's maximum and is starting over) is fed to the input of the next one, then the resolution will:

a) increase b) decrease c) stay the same


Z_REG = Z_IN
IN1 <= IN2
IN1 > IN2
Z_REG = Z_REG / 2
Z_REG = Z_REG * Z_REG
Z_REG > 5
Z_REG <= 5
OUT_REG = Z_REG

A few midterm solutions…



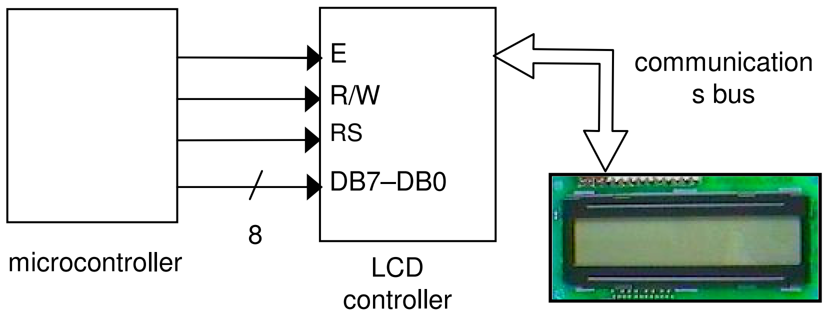


If signal is not declared, assume it is zero.

Peripherals continued…

We now continue where we left off before the midterm. Peripherals (standard single purpose processors) from chapter 4 in Embedded System Design.

LCD controller



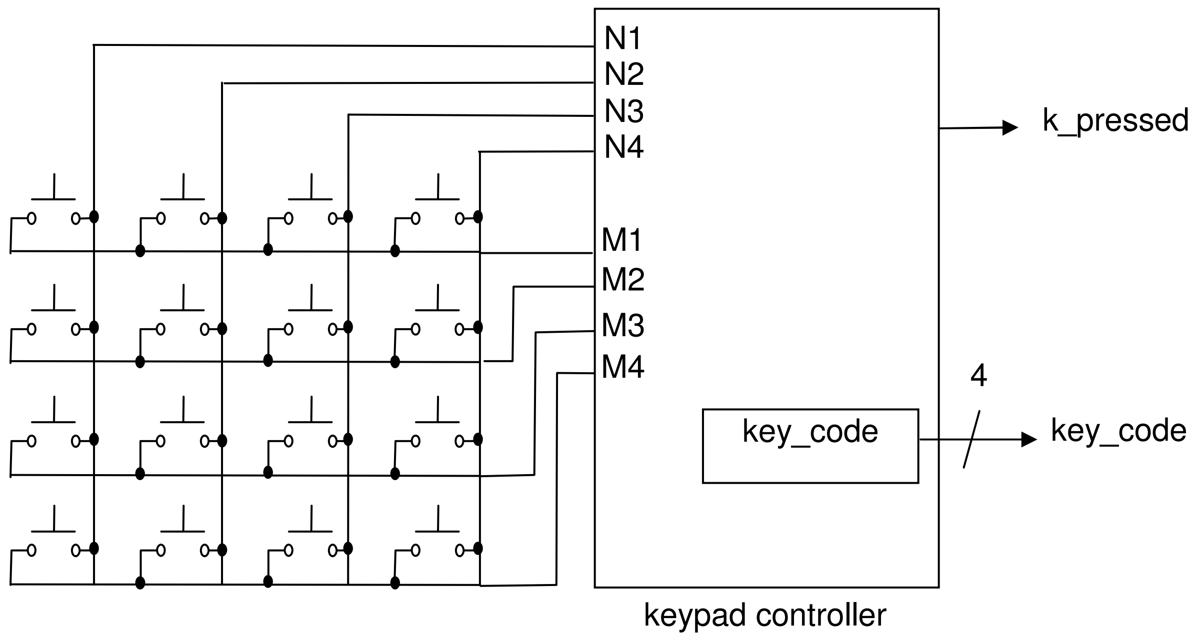

```
void WriteChar(char c){

   RS = 1;                          /* indicate data being sent */
   DATA_BUS = c;                  /* send data to LCD */
   EnableLCD(45);                  /* toggle the LCD with appropriate
delay */
}
```

CODES	
I/D = 1 cursor moves left	DL = 1 8-bit
I/D = 0 cursor moves right	DL = 0 4-bit
S = 1 with display shift	N = 1 2 rows
S/C =1 display shift	N = 0 1 row
S/C = 0 cursor movement	F = 1 5x10 dots
R/L = 1 shift to right	F = 0 5x7 dots
R/L = 0 shift to left	

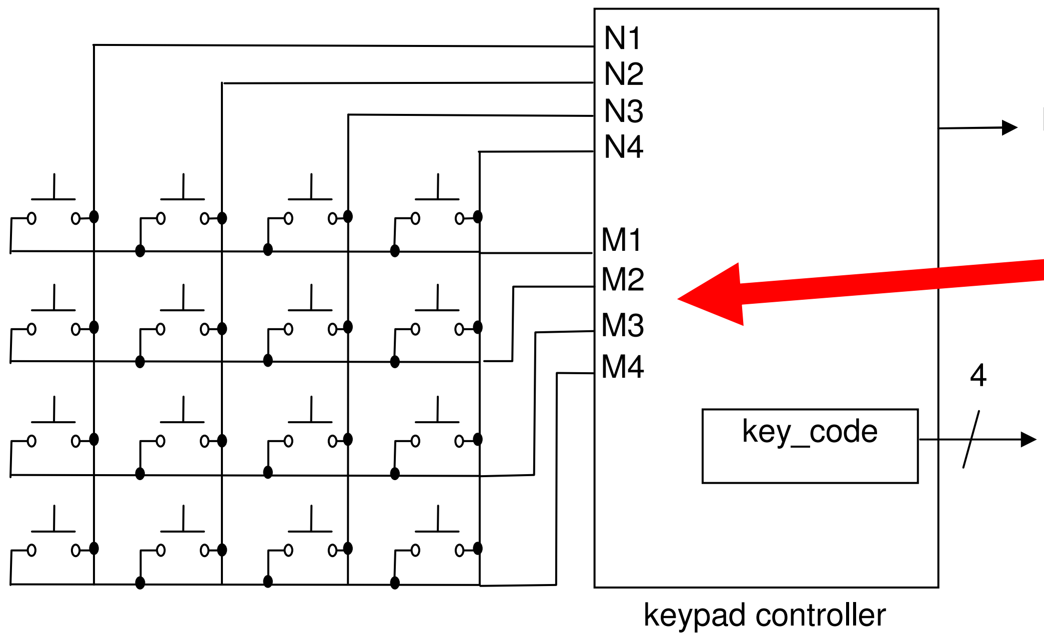
RS	R/W	DB_7	DB_6	DB_5	DB_4	DB_3	DB_2	DB_1	DB_0	Description
0	0	0	0	0	0	0	0	0	1	Clears all display, return cursor home
0	0	0	0	0	0	0	0	1	*	Returns cursor home
0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and/or specifies not to shift display
0	0	0	0	0	0	1	D	C	B	ON/OFF of all display(D), cursor ON/OFF (C), and blink position (B)
0	0	0	0	0	1	S/C	R/L	*	*	Move cursor and shifts display
0	0	0	0	1	DL	N	F	*	*	Sets interface data length, number of display lines, and character font
1	0	WRITE DATA								Writes Data

Keypad controller



N=4, M=4

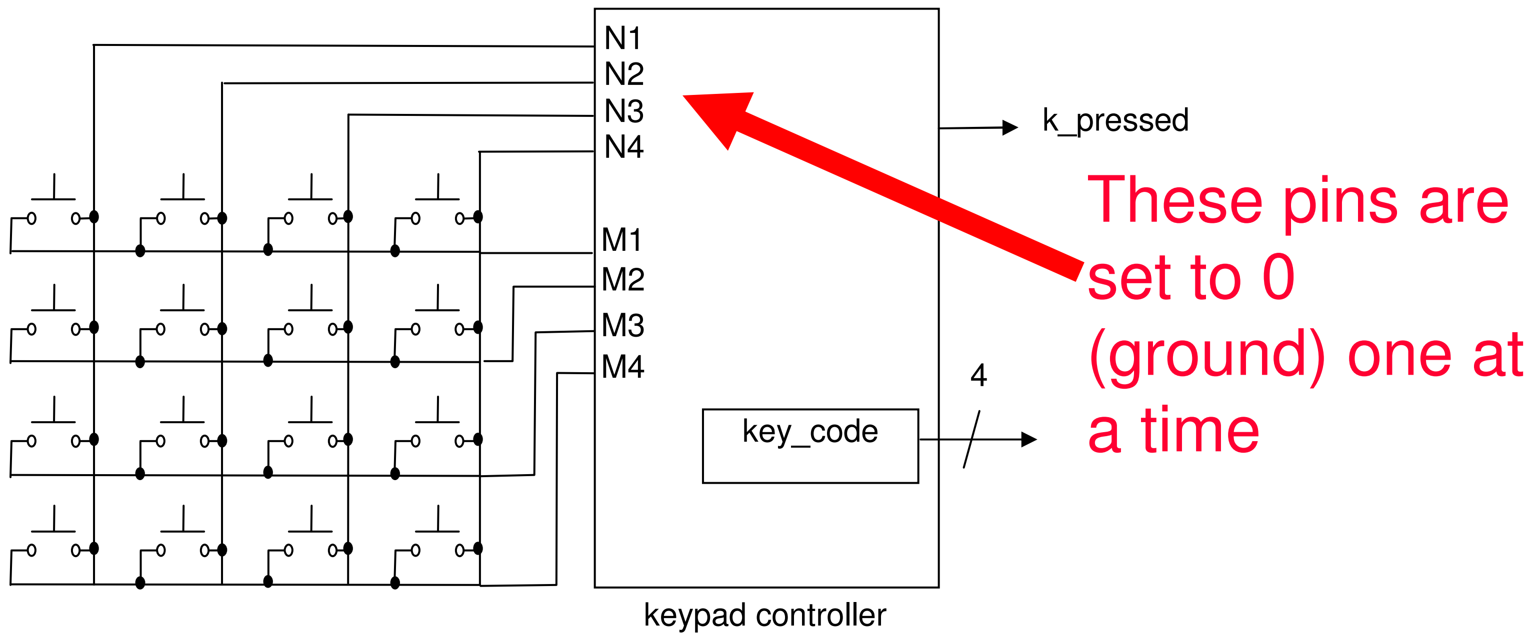
Keypad controller



These all are weakly pulled high so they equal 1 when not connected to ground.

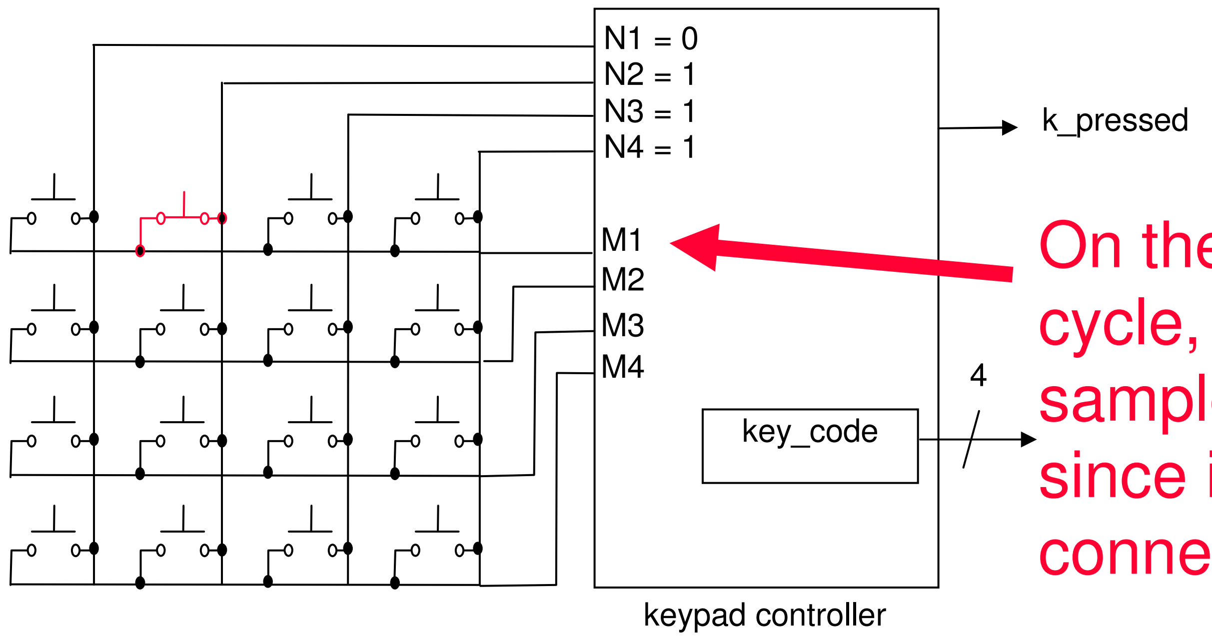
N=4, M=4

Keypad controller



N=4, M=4

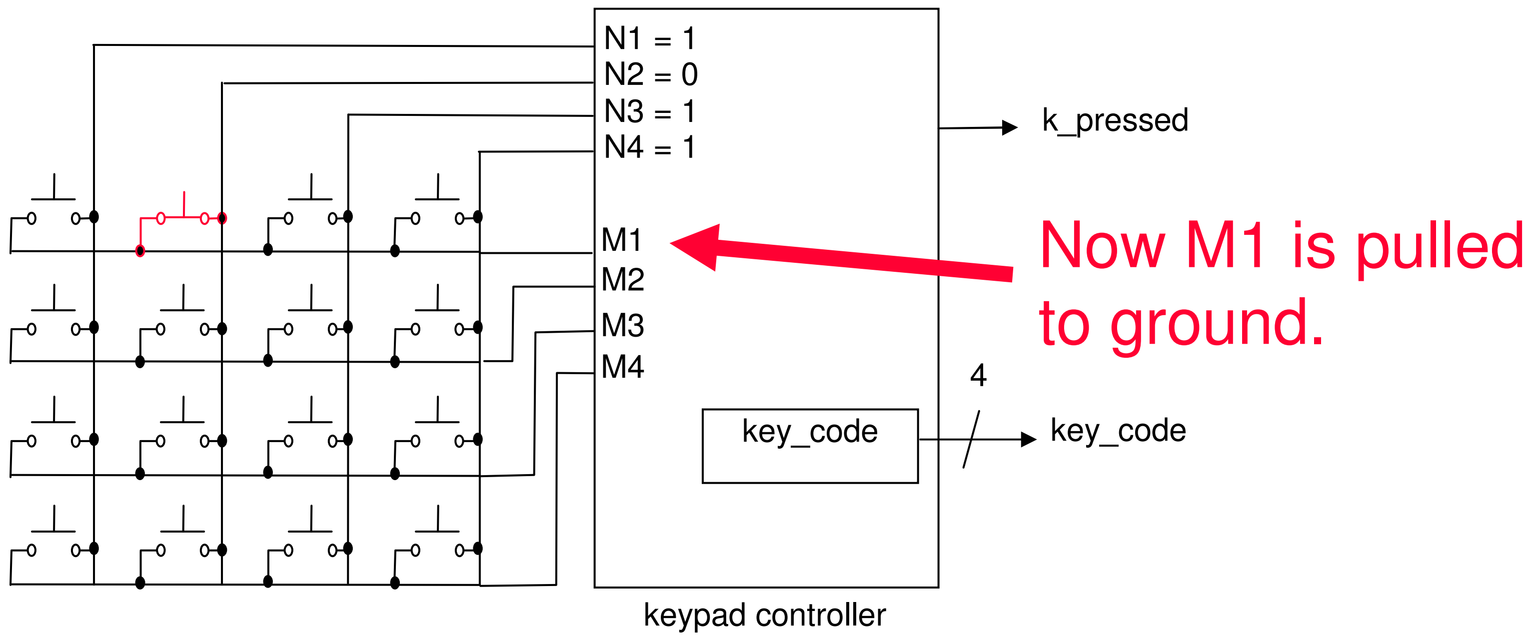
Keypad controller



On the first cycle, M1 is sampled high, since it is connected to N2.

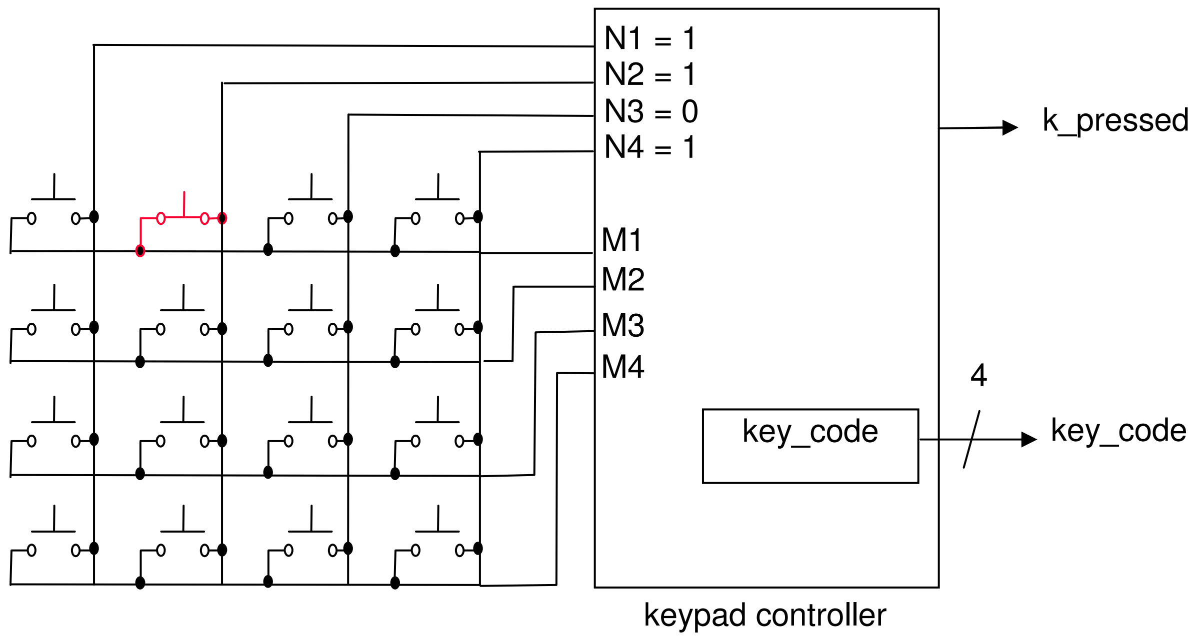
N=4, M=4

Keypad controller



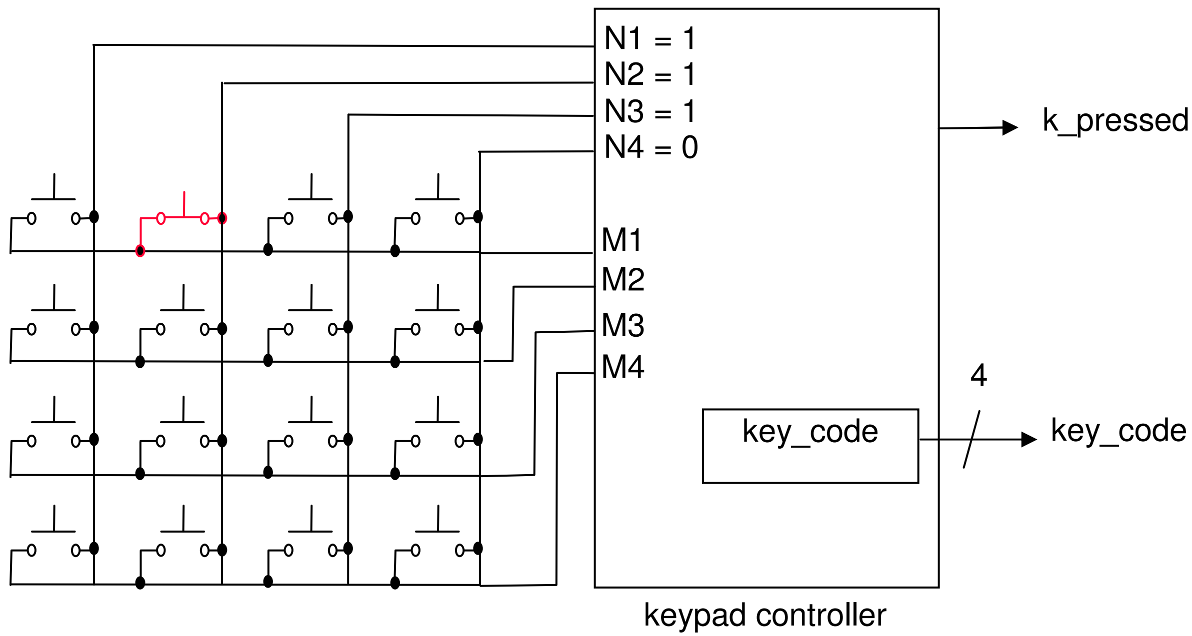
N=4, M=4

Keypad controller



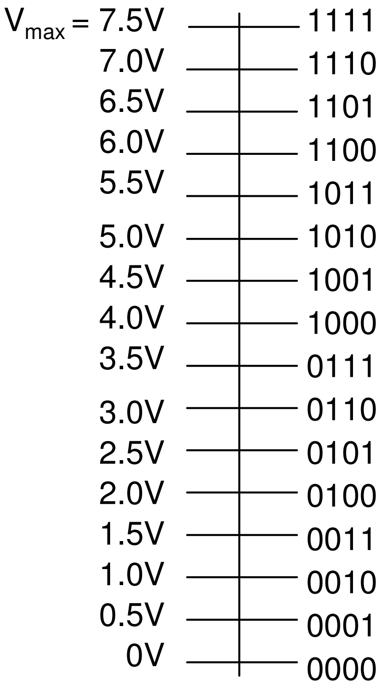
N=4, M=4

Keypad controller



N=4, M=4

Analog-to-digital converters



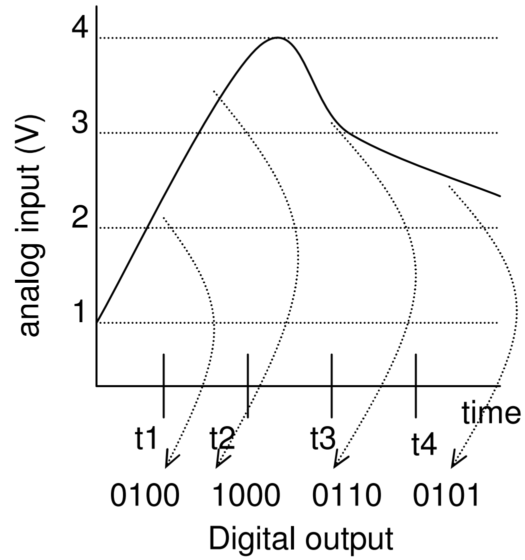
proportionality

Assume your range starts at zero, then:

$$\mathbf{e/V_{max} = d/(2^n - 1)}$$

Where 'e' is the analog value, and d is the digital value.

If your range does not start at zero, shift 'e' and V_{max} accordingly.



analog to digital

Analog to Digital conversion using successive approximation

Given an analog input signal whose voltage should range from 0 to 15 volts, and an 8-bit digital encoding, calculate the correct encoding for 5 volts. Then trace the successive-approximation approach to find the correct encoding.

$5/15 = d/(2^8-1)$

$d = 85$

Encoding: 01010101

Successive-approximation method

Step	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
½(V_{max} – V_{min}) = 7.5 volts V_{max} = 7.5 volts.	0	0	0	0	0	0	0	0
½(7.5 + 0) = 3.75 volts V_{min} = 3.75 volts.	0	1	0	0	0	0	0	0
½(7.5 + 3.75) = 5.63 volts V_{max} = 5.63 volts	0	1	0	0	0	0	0	0
½(5.63 + 3.75) = 4.69 volts V_{min} = 4.69 volts.	0	1	0	1	0	0	0	0
½(5.63 + 4.69) = 5.16 volts V_{max} = 5.16 volts.	0	1	0	1	0	0	0	0
½(5.16 + 4.69) = 4.93 volts V_{min} = 4.93 volts.	0	1	0	1	0	1	0	0
½(5.16 + 4.93) = 5.05 volts V_{max} = 5.05 volts.	0	1	0	1	0	1	0	0
½(5.05 + 4.93) = 4.99 volts	0	1	0	1	0	1	0	1

Analog-to-digital converters

Successive Approximation gives you an idea of how one type of A/D (sigma delta) works, but the easiest way to calculate values is to use:

$$e/V_{max} = d/(2^n - 1)$$

Example: You have a 12 bit A/D and the range of the A/D is -1V to 1V. If the output of the A/D is: "001010111111" what analog voltage is at the input?

Analog-to-digital converters

Example: You have a 12 bit A/D and the range of the A/D is -1V to 1V. If the output of the A/D is: “001010111111” what analog voltage is at the input?

We are trying to determine the analog input which is **e**.

$e/V_{max} = d/(2^n – 1)$

$e/(1 – (-1)) = d/(2^n – 1)$; *note: the range is shifted by 1V*

d = “001010111111” = 703; n = 12; $2^n = 4096$;

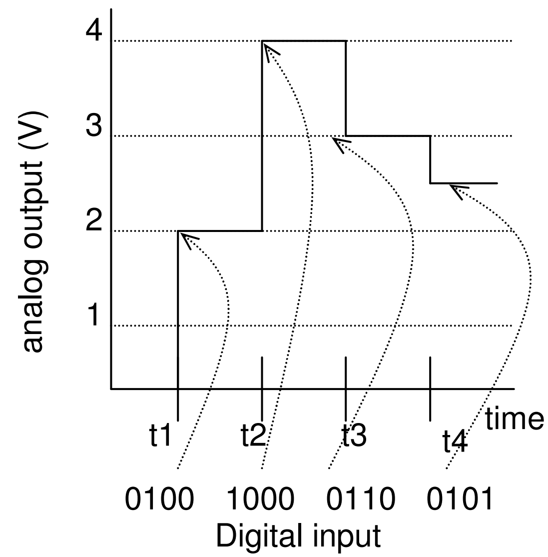
e/2 = 703/(4096 – 1) = 703/4095 = 0.1716722

e = 0.343345; shifted back e = -1 + 0.343345

= -0.656654

Digital to Analog converters

- Quite Simply, the go the opposite direction of ADC's.
- You send it a digital value, and it produces an analog voltage proportional to that value.
- Among other things, DAC's are critical to audio components that store the media in digital format.



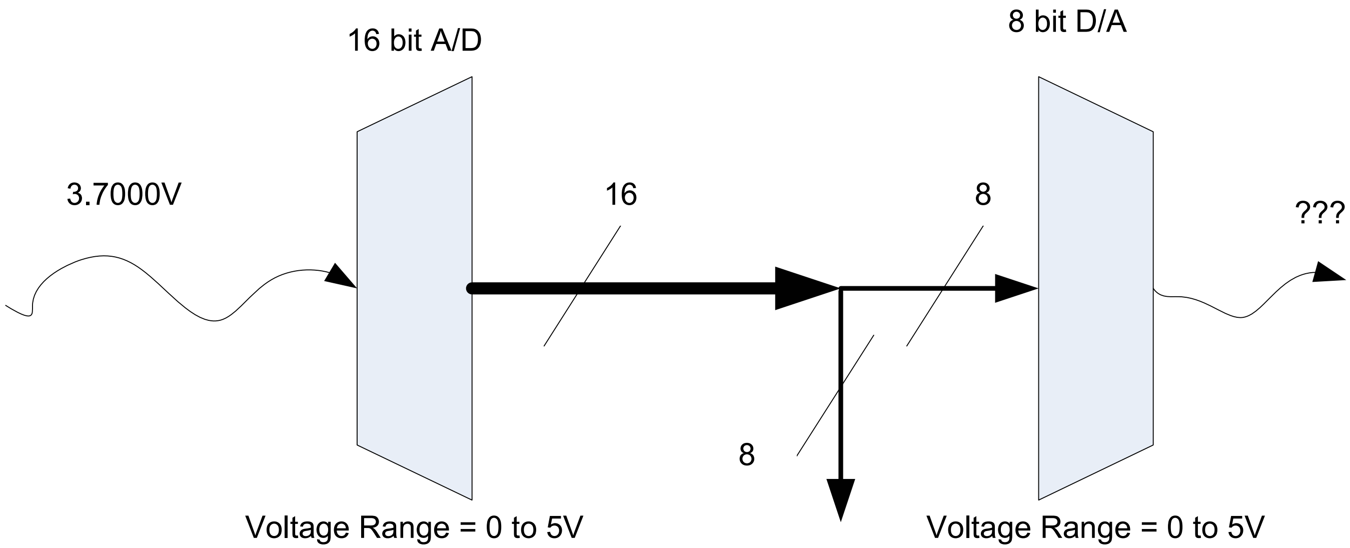
digital to analog

In-Class Exercises

- You have a 4-bit ADC which has an input range from 2V to 3V. It is sending the value “1110.” What is the analog input?
- Can you determine the tolerance (ie: accuracy) of you answer?

In-Class Exercises

- A 16 bit A/D with a 0 to 5V range has it's 8 most significant bits connected to an 8 bit D/A with a range of 0 to 5V. If the input to the A/D is 3.700V what is the output of the D/A?



In-Class Exercises

- You have a 4-bit ADC which has an input range from 2V to 3V. It is sending the value “1110.” What is the analog input? Can you determine the tolerance (ie: accuracy) of you answer?

e/V = d(2^n-1)   e/(3-2) = 14/15 e=0.93;

shifted back **e = 2.93V**

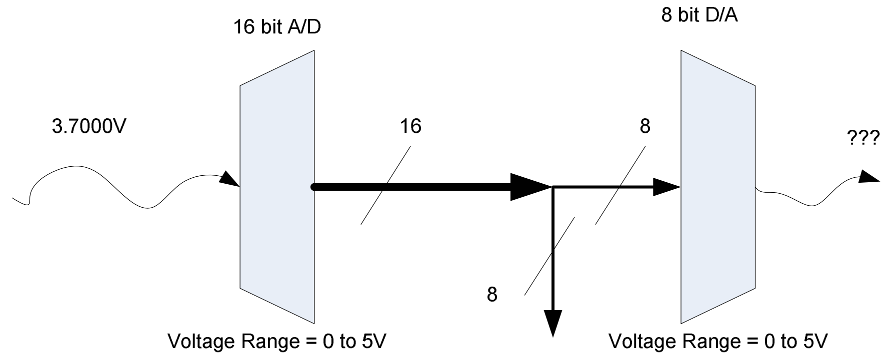
The tolerance is basically the +/- ½(resolution)

The resolution is the smallest change the A/D can measure (or *resolve*—get it?)

For this example the resolution is 1/15 * 1 = 0.067V (smallest digital step * range)

So, the accuracy is **+/- 0.034V**

In-Class Exercises



for the A/D: 3.7000/5 = d/(65535), so d=48496.64≈48497

48497 = “1011110101110001”

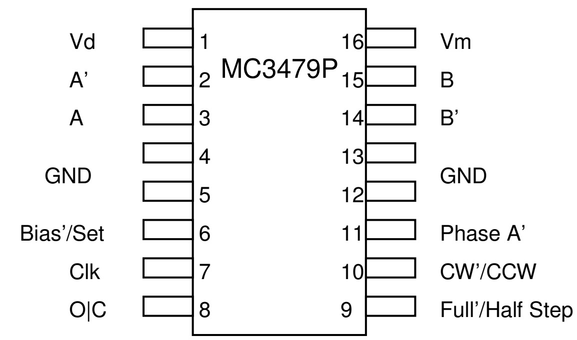
for the D/A: e/5 = “10111101”/255 = 189/255 = .74117

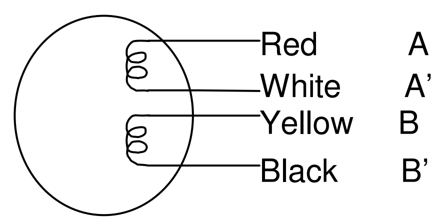
e = 3.7058V

Stepper motor controller

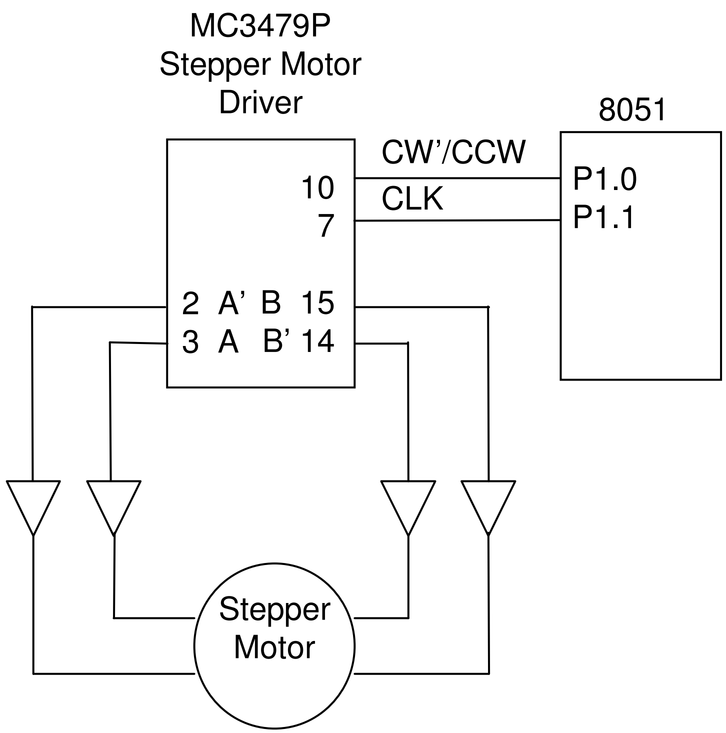
- Stepper motor: rotates fixed number of degrees when given a "step" signal
  - In contrast, DC motor just rotates when power applied, coasts to stop
- Rotation achieved by applying specific voltage sequence to coils
- Controller greatly simplifies this

Sequence	A	B	A'	B'
1	+	+	-	-
2	-	+	+	-
3	-	-	+	+
4	+	-	-	+
5	+	+	-	-





Stepper motor with controller (driver)




```
/* main.c */

sbit clk=P1^1;
sbit cw=P1^0;

void delay(void){
  int i, j;
  for (i=0; i<1000; i++)
    for ( j=0; j<50; j++)
      i = i + 0;
}
```

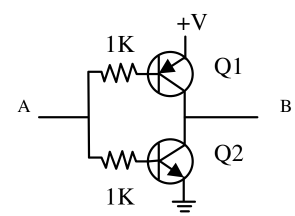
```
void main(void){

  */turn the motor forward */
  cw=0;              /* set direction
*/
  clk=0;             /* pulse clock */
  delay();
  clk=1;

  /*turn the motor backwards */
  cw=1;              /* set direction
*/
  clk=0;             /* pulse clock */
  delay();
  clk=1;

}
```

The output pins on the stepper motor driver do not provide enough current to drive the stepper motor. To amplify the current, a buffer is needed. One possible implementation of the buffers is pictured to the left. Q1 is an MJE3055T NPN transistor and Q2 is an MJE2955T PNP transistor. A is connected to the 8051 microcontroller and B is connected to the stepper motor.



Real Time Clocks (RTC)

- Basically, it is the systems wrist-watch
- Typically, the keep seconds, minutes, hours, days, months, years, and some times centuries.
- Should account for leap-years.
- Most of the time, the system communicates with the RTC on a serial bus. Either setting the time, or requesting the time.
- Naturally, it is going to need some sort of battery back-up, or get set every time it powers up.

Things to do before Thursday…

- Read through page 123 in ESD (The beginning of Memory)