

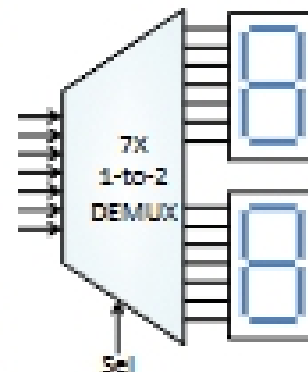
Lecture 9: Project 1 discussion

Project 1 tasks

- For the alarm clock, you need to develop your own functions to
 - show all four digits on two SSDs simultaneously
 - drive outputs based on current states
 - determine next states based on inputs and current states

Seven Segment Display

- Each PmodSSD is able to display two digits at the same time.
- For each digit, seven inputs are required to control all the seven segments.
- So it seems that fourteen inputs are needed for displaying two digits.
- Yet there are only 8 inputs available for each PmodSSD (despite GND and VCC)
 - Depending on the value of "Sel", inputs are sent to different 7-seg LEDs receive inputs



Q: Using these limited 8 inputs, how to display both digits at the same time?

How to display one digit?

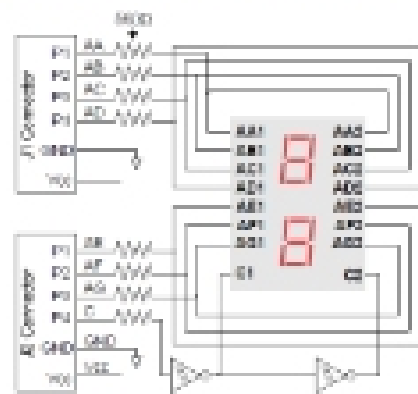
First, define the port bits

```
// 7 Segment Display pmod for using the TOP JA & JB jumpers
// Segments
#define SegA  PORTEbits.RE0
#define SegB  PORTEbits.RE1
#define SegC  PORTEbits.RE2
#define SegD  PORTEbits.RE3
#define SegE  PORTGbits.RG9
#define SegF  PORTGbits.RG8
#define SegG  PORTGbits.RG7
// Display selection. 0 = right, 1 = left
#define DispSel PORTGbits.RG6
```

How did we figure this out?

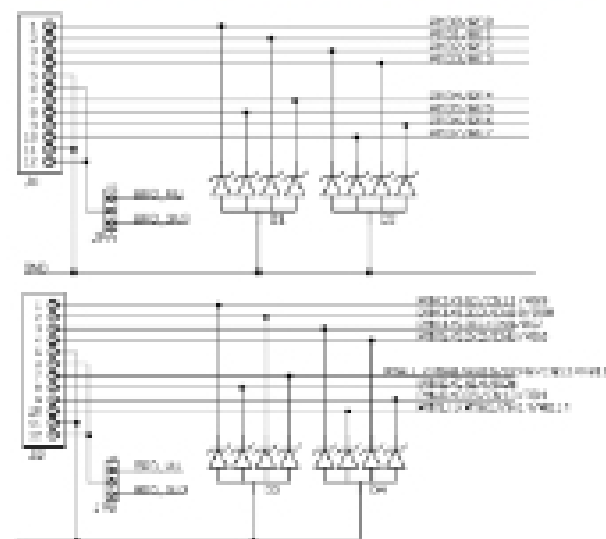
Port definitions

From Pmod SSD_rm.pdf



Seven-Segment Display Connection Diagram

From Cerebot 32MX4_sch_D.pdf



1/28/2014

CPE6010-ug10-4a-cb

7

How to display one digit?

Then, develop a function to drive those bits based on inputs

```
void displayDigit (unsigned char sel, unsigned char value){
    DispSel = sel;
    SegA = value & 1;
    SegB = (value >> 1) & 1;
    SegC = (value >> 2) & 1;
    SegD = (value >> 3) & 1;
    SegE = (value >> 4) & 1;
    SegF = (value >> 5) & 1;
    SegG = (value >> 6) & 1;
}
```

1/28/2014

CPE6010-ug10-4a-cb

8

Use an array to store display vectors

// look-up table for the display vectors

```
unsigned char number[]={
    0b0111111, //0
    0b0000110, //1
    0b1011011, //2
    0b1001111, //3
    0b1100110, //4
    0b1101101, //5
    0b1111101, //6
    0b0000111, //7
    0b1111111, //8
    0b1101111, //9
    0b0000000 //clear
};
```

1/28/2014

CPE6010-ug10-4a-cb

7

How to show 2 digits simultaneously?

Periodically refresh!

```
void showNumber(int display_value){
    int i;
    displayDigit ( 0, number[display_value%10] );
    // refresh
    for (i = 0; i < REFRESH_RATE; i++);

    displayDigit ( 1, number[display_value/10] );
    // refresh
    for (i = 0; i < REFRESH_RATE; i++);
}
```

Q: What should the value of REFRESH_RATE be?

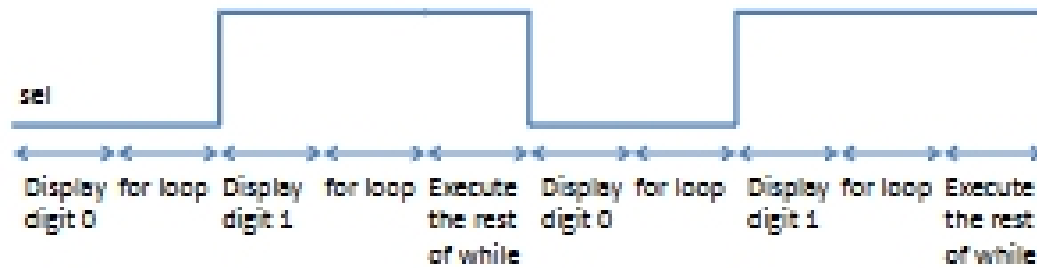
1/28/2014

CPE6010-ug10-4a-cb

8

How to show 2 digits simultaneously?

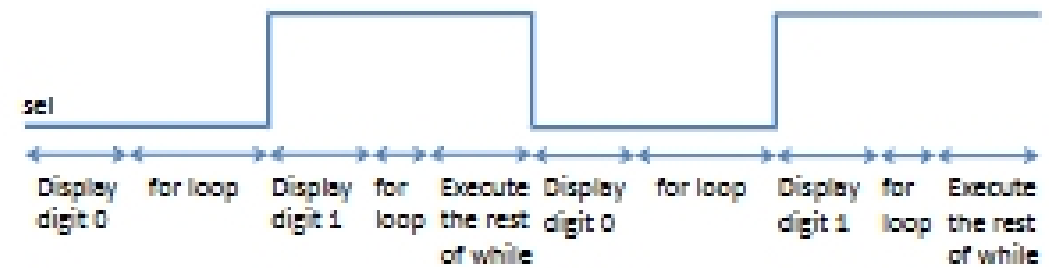
- The value of REFRESH_RATE determines refresh frequency
 - Too slow: can observe refresh
 - Too fast: LED not bright enough



One digit will be brighter than the other!

How to show 2 digits simultaneously?

- The value of REFRESH_RATE determines refresh frequency
 - Too slow: can observe refresh
 - Too fast: LED not bright enough



More balanced brightness

Project 1 tasks

- For the alarm clock, you need to develop your own functions to
 - show all four digits on two SSDs simultaneously
 - drive outputs based on current states
 - determine next states based on inputs and current states

Well-Structured Code

Decoupling output and next state logic!

```
int locked = 0;
while(1) {
    // output logic
    switch (mode) {
    case A:
        -
    case B:
        -
    }
    // next state logic
    if (Btn1) {
        -
    }
    else if (Btn2) {
        -
    }
    else if (!Btn1 && !Btn2) {
        -
    }
}
```

Handle current iteration

Outputs depend on current state but not inputs – *Moore machine*

Prepare for next iteration

Next states evaluated only when inputs change – *Event driven*

Next states become current states

Simpler to code, easier to debug!