/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Author: Group 2 (Hang Xu, Wen Wu, Wenjun Ma)

Date complete: 13/11/2017

Filename: EE2A Experiment2 PIC Introduction

Target device: PIC18F27K40

Fuse settings: default

Program function:

(i)To use IO-mapped data structures to talk to interface pins.

(ii)To implement the basic functionality of a synchronous finite state machine and the equivalent implementation on a processor.

(iii)To combine multiple, disjoint program fragments together into one package that can be rapidly demonstrated.

(iv)To implement logical bit-wise operations in software

(v)To implement look-up tables on both inputs and outputs (lambda logic) of synthesised finite state machines.

Testing video:

https://1drv.ms/v/s!AlOd1b5oIKMagcgCAGvCIfotrDwoCw

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<18F27K40.h>

#include<stdio.h>

#use delay(internal=64MHZ,clock\_out)

void test() //test hardware

{

while(TRUE)

{

output\_high(PIN\_C0);//flashing at 1hz

delay\_ms(500);

output\_low(PIN\_C0);

delay\_ms(500);

}

}

void sdb()//set frequency back to 64Mhz

{

setup\_oscillator(0x00000860);//64Mhz

setup\_adc\_ports(NO\_ANALOGS);

}

void s0() //main frequency=4Mhz

{

int clocksignal; //clock signal

int freq = 2; //clock frequency

int counter = 1; //counter

int remainder; //remainder

setup\_oscillator(0x00000260);// set main frequency at 4MHz

setup\_adc\_ports(NO\_ANALOGS);

while(TRUE)

{

//judge part1

remainder = counter%2; //complementation

if(remainder == 1)

{

output\_high(PIN\_C0);

}

if(remainder == 0)

{

output\_low(PIN\_C0);

}

//clock part1

clocksignal = 1;//clocksignal is high

counter = counter + 1;

delay\_ms(1000/(32\*freq));

clocksignal = 0;//clocksignal is low

delay\_ms(1000/(32\*freq));

//judge part2

remainder = counter%2;

if(remainder == 1)

{

output\_high(PIN\_C0);

}

if(remainder == 0)

{

output\_low(PIN\_C0);

}

//clock part2

clocksignal = 1;//clocksignal is high

counter = counter + 1;

delay\_ms(1000/(32\*freq));

clocksignal = 0;//clocksignal is low

delay\_ms(1000/(32\*freq));

if(!((input(PIN\_B4)==0)&&(input(PIN\_B5)==0)&&(input(PIN\_B6)==0)&&(input(PIN\_B7)==0))) //if input no longer equals 0000, break S0 and return

{

return;

}

}

}

void s1() //main frequency = 8Mhz

{

int clocksignal; //clock signal

int freq = 2; //clock frequency

int counter = 1; //counter

int remainder; //remainder of counter divided by 2

setup\_oscillator(0x00000360);//set main frequency at 8Mhz

setup\_adc\_ports(NO\_ANALOGS);

while(TRUE)

{

//judge part1

remainder = counter%2;

if(remainder == 1)

{

output\_high(PIN\_C0);

}

if(remainder == 0)

{

output\_low(PIN\_C0);

}

//clock part1

clocksignal = 1;//clocksignal is high

counter = counter + 1;

delay\_ms(1000/(16\*freq));

clocksignal = 0;//clocksignal is low

delay\_ms(1000/(16\*freq));

//judge part2

remainder = counter%2;

if(remainder == 1)

{

output\_high(PIN\_C0);

}

if(remainder == 0)

{

output\_low(PIN\_C0);

}

//clock part2

clocksignal = 1;//clocksignal is high

counter = counter + 1;

delay\_ms(1000/(16\*freq));

clocksignal = 0;//clocksignal is low

delay\_ms(1000/(16\*freq));

if(!((input(PIN\_B4)==0)&&(input(PIN\_B5)==0)&&(input(PIN\_B6)==0)&&(input(PIN\_B7)==1))) //if input no longer equals 0001, break S1 and return

{

return;

}

}

}

void s2()//main frequency=16MHz

{

int clocksignal; //clock signal

int freq = 2; //clock frequency

int counter = 1; //counter

int remainder; //remainder

setup\_oscillator(0x00000560);//set main frequency at 16MHz

setup\_adc\_ports(NO\_ANALOGS);

while(TRUE)

{

//judge part1

remainder = counter%2;

if(remainder == 1)

{

output\_high(PIN\_C0);

}

if(remainder == 0)

{

output\_low(PIN\_C0);

}

//clock part1

clocksignal = 1;

counter = counter + 1;

delay\_ms(1000/(8\*freq));

clocksignal = 0;

delay\_ms(1000/(8\*freq));

//judge part2

remainder = counter%2;

if(remainder == 1)

{

output\_high(PIN\_C0);

}

if(remainder == 0)

{

output\_low(PIN\_C0);

}

//clock part

clocksignal = 1;

counter = counter + 1;

delay\_ms(1000/(8\*freq));

clocksignal = 0;

delay\_ms(1000/(8\*freq));

if(!((input(PIN\_B4)==0)&&(input(PIN\_B5)==0)&&(input(PIN\_B6)==1)&&(input(PIN\_B7)==0)))//if input no longer equals 0010, break S2 and return

{

return;

}

}

}

void s3()// T-type flip flop in toggle mode. Pin RC0 produces 1 Hz

{

while(TRUE)

{

s0();//call S0, main frequency = 4MHz

s1();//call S1, main frequency = 8MHz

s2();//call S2, main frequency = 16MHz

if(!((input(PIN\_B4)==0)&&(input(PIN\_B5)==0)&&(input(PIN\_B6)==1)&&(input(PIN\_B7)==1)))//if input no longer equals 0011, break S3 and return

{

return;

}

}

}

void s4()//Port C0..5 implements a 6-bit pure-binary up-counter

{

int counter = 0;//count from 0

int freq = 1;

sdb();

while(TRUE)

{

if(counter==64)//if already count to 64, reset count to 0

{

counter=0;

}

output\_c(counter);

delay\_ms(1000/freq);

counter=counter+1;

if(!((input(PIN\_B4)==0)&&(input(PIN\_B5)==1)&&(input(PIN\_B6)==0)&&(input(PIN\_B7)==0)))//if input no longer equals 0100, break S4 and return

{

output\_c(0);

return;

}

}

}

void s5()//Port C0..5 implements a 6-bit pure-binary down-counter

{

int counter = 63;

int freq = 1;

sdb();

while(TRUE)

{

if(counter==-1)//if have already counted to 0, reset to 63

{

counter=63;

}

output\_c(counter);

delay\_ms(1000/freq);

counter=counter-1;

if(!((input(PIN\_B4)==0)&&(input(PIN\_B5)==1)&&(input(PIN\_B6)==0)&&(input(PIN\_B7)==1)))//if input no longer equals 0101, break S5 and return

{

output\_c(0);

return;

}

}

}

void s6()//6-bit Gray-code up-counter

{

int counter = 0;

int freq = 1;

int a,b,c,d,e,f;//value of each bit of counter a=LSB

int a1,b1,c1,d1,e1,f1;//intermediate variable a1:C0,b1:C1...

int decimal;

sdb();

while(TRUE)

{

if(counter==64)//if already counter to 64,reset to 0

{

counter = 0;

}

// take out the value of each bit of counter

a = counter%2;// LSB, output C0

b = ((counter - a)/2)%2;// output C1

c = ((((counter - a)/2) - b)/2)%2;// output C2

d = ((((((counter - a)/2) - b)/2) - counter)/2)%2;// output C3

e = ((((((((counter - a)/2) - b)/2) - counter)/2) - d)/2)%2;// output C4

f = ((((((((((counter - a)/2) - b)/2) - counter)/2) - d)/2) - e)/2)%2;// MSB output C5

// calculate gray code

f1 = f;

e1 = f + e;

if(e1 == 2){e1 = 0;}

d1 = e + d;

if(d1 == 2){d1 = 0;}

c1 = d + c;

if(c1 == 2){c1 = 0;}

b1 = c + b;

if(b1 == 2){b1 = 0;}

a1 = b + a;

if(a1 == 2){a1 = 0;}

// set the value of intermediate variables to output variables

a = a1;

b = b1;

c = c1;

d = d1;

e = e1;

f = f1;

// transfer gray code into decimal value to output

decimal = 32\*f + 16\*e + 8\*d + 4\*c + 2\*b + a ;

output\_c(decimal);

delay\_ms(1000/freq);

counter=counter+1;

if(!((input(PIN\_B4)==0)&&(input(PIN\_B5)==1)&&(input(PIN\_B6)==1)&&(input(PIN\_B7)==0)))//if input no longer equals 0110, break S6 and return

{

output\_c(0);

return;

}

}

}

void s7() //3-bit pure-binary up-counter and 3-bit Gray-code up-counter

{

int counter = 0;

int freq = 1;

int a,b,c;//value of each bit of counter a=LSB

int a1,b1,c1;//intermediate variable a1:C0,b1:C1...

int decimalvalue;//decimal value to output

sdb();

while(TRUE)

{

if(counter==8)

{

counter=0;//reset

}

a = counter%2;//take out value of each bit of counter

b = ((counter - a)/2)%2;

c = ((((counter - a)/2) - b)/2)%2;

// calculate gray code

c1 = c;

b1 = c + b;

if(b1 == 2){b1 = 0;}

a1 = b + a;

if(a1 == 2){a1 = 0;}

// set the value of intermediate variables to output variables

a = a1;

b = b1;

c = c1;

// transfer gray code into decimal value to output

decimalvalue = counter + 32\*c + 16\*b + 8\*a ;

output\_c(decimalvalue);

delay\_ms(1000/freq);

counter=counter+1;

if(!((input(PIN\_B4)==0)&&(input(PIN\_B5)==1)&&(input(PIN\_B6)==1)&&(input(PIN\_B7)==1)))//if input no longer equals 0111, break S7 and return

{

output\_c(0);

return;

}

}

}

void s8()//bitwise AND operation, input B0B1,B2B3,outputC0,C1

{

sdb();

while(TRUE)

{

if(input(PIN\_B0)\*input(PIN\_B2)==1)//if B0\*B2=1, outputC0=1

{

output\_high(PIN\_C0);

}

else

{

output\_low(PIN\_C0);

}

if(input(PIN\_B1)\*input(PIN\_B3)==1)//ifB1\*B3=1,outputC1=1

{

output\_high(PIN\_C1);

}

else

{

output\_low(PIN\_C1);

}

if(!((input(PIN\_B4)==1)&&(input(PIN\_B5)==0)&&(input(PIN\_B6)==0)&&(input(PIN\_B7)==0)))//if input no longer equals 1000, break S8 and return

{

output\_c(0);

return;

}

}

}

void s9()//bitwise OR operation, input B0B1,B2B3,outputC0,C1

{

sdb();

while(TRUE)

{

if(input(PIN\_B0)+input(PIN\_B2)==0)//if B0+B2=0, outputC0=0

{

output\_low(PIN\_C0);

}

else//if B0+B2 not equals 0, outputC0=1

{

output\_high(PIN\_C0);

}

if(input(PIN\_B1)+input(PIN\_B3)==0)//if B1+B3=0, outputC1=0

{

output\_low(PIN\_C1);

}

else//if B1+B3 not equals 0, outputC1=1

{

output\_high(PIN\_C1);

}

if(!((input(PIN\_B4)==1)&&(input(PIN\_B5)==0)&&(input(PIN\_B6)==0)&&(input(PIN\_B7)==1)))//if input no longer equals 1001, break S9 and return

{

output\_c(0);

return;

}

}

}

void s10()//bitwise NOT operation

{

sdb();

while(TRUE)

{

if(input(PIN\_B0)==1)//if input=1, output=0

{

output\_low(PIN\_C0);

}

else

{

output\_high(PIN\_C0); // if input=0, output = 1

}

if(input(PIN\_B1)==1)

{

output\_low(PIN\_C1);

}

else

{

output\_high(PIN\_C1);

}

if(input(PIN\_B2)==1)

{

output\_low(PIN\_C2);

}

else

{

output\_high(PIN\_C2);

}

if(input(PIN\_B3)==1)

{

output\_low(PIN\_C3);

}

else

{

output\_high(PIN\_C3);

}

if(!((input(PIN\_B4)==1)&&(input(PIN\_B5)==0)&&(input(PIN\_B6)==1)&&(input(PIN\_B7)==0)))//if input no longer equals 1010, break S10 and return

{

output\_c(0);

return;

}

}

}

void s11()//bitwise Exclusive-OR operation, input B0B1,B2B3,outputC0,C1

{

sdb();

while(TRUE)

{

if(input(PIN\_B0)==input(PIN\_B2)) //if input1=input2, output=0

{

output\_low(PIN\_C0);

}

else////if input1!=input2, output=1

{

output\_high(PIN\_C0);

}

if(input(PIN\_B1)==input(PIN\_B3))

{

output\_low(PIN\_C1);

}

else

{

output\_high(PIN\_C1);

}

if(!((input(PIN\_B4)==1)&&(input(PIN\_B5)==0)&&(input(PIN\_B6)==1)&&(input(PIN\_B7)==1)))//if input no longer equals 1011, break S11 and return

{

output\_c(0);

return;

}

}

}

Void s12()//set-reset flip-flop, B0=set,B1=reset,C0=output

{

sdb();

while(TRUE)

{

//if((input(PIN\_B0)==0)&&(input(PIN\_B1)==0))// if set=reset=0, keep

//{

// output\_low(PIN\_C0);

//}

if((input(PIN\_B0)==0)&&(input(PIN\_B1)==1))//if set=0, reset=1, output=0

{

output\_low(PIN\_C0);

}

if((input(PIN\_B0)==1)&&(input(PIN\_B1)==0))//if set=1,reset=0,output=1

{

output\_high(PIN\_C0);

}

if((input(PIN\_B0)==1)&&(input(PIN\_B1)==1))//if set=1, reset=1, output= flashing light

{

output\_high(PIN\_C0);

delay\_ms(500);

output\_low(PIN\_C0);

delay\_ms(500);

}

if(!((input(PIN\_B4)==1)&&(input(PIN\_B5)==1)&&(input(PIN\_B6)==0)&&(input(PIN\_B7)==0)))//if input no longer equals 1100, break S12 and return

{

output\_c(0);

return;

}

}

}

void s13()//input look-up table

{

sdb();

int freq;

int delaytime;

while(true)

{

int i=input(PIN\_B3)+(input(PIN\_B2)\*2)+(input(PIN\_B1)\*4)+(input(PIN\_B0)\*8);//B0=MSB i is the decimal value of input

if (i == 0){freq=2;}//frequecy change according to input

if (i == 1){freq=4;}

if (i == 2){freq=5;}

if (i == 3){freq=6;}

if (i == 4){freq=7;}

if (i == 5){freq=8;}

if (i == 6){freq=9;}

if (i == 7){freq=10;}

if (i == 8){freq=20;}

if (i == 9){freq=21;}

if (i == 10){freq=22;}

if (i == 11){freq=23;}

if (i == 12){freq=24;}

if (i == 13){freq=27;}

if (i == 14){freq=28;}

if (i == 15){freq=30;}

delaytime = 1000/freq;

output\_c(1);

delay\_ms(delaytime);//flashing light

output\_c(2);

delay\_ms(delaytime);

output\_c(4);

delay\_ms(delaytime);

output\_c(8);

delay\_ms(delaytime);

output\_c(16);

delay\_ms(delaytime);

output\_c(32);

delay\_ms(delaytime);

output\_c(64);

delay\_ms(delaytime);

output\_c(128);

delay\_ms(delaytime);

output\_c(0);

if(!((input(PIN\_B4)==1)&&(input(PIN\_B5)==1)&&(input(PIN\_B6)==0)&&(input(PIN\_B7)==1)))//if input no longer equals 1101, break S13 and return

{

output\_c(0);

return;

}

}

}

void s14()//bright-eyed and bushy tailed capability

{

sdb();

float num=1.0;

int sp;

while(true)

{

//edge judge part

if(num==384){num=129;}

if(num==258){num=3;}

if(num==1.5){num=129;}

if(num==64.5){num=192;}

if((num==3)||(num==6)||(num==12)||(num==24)||(num==48)||(num==96)||(num==192)||(num==384)){num=(num/3);}

if(num==129){num=128;}

if((num==1)||(num==2)||(num==4)||(num==8)||(num==16)||(num==32)||(num==64)||(num==128)||(num==256)||(num==0.5)){num=num;}

while(input(PIN\_B2)==1&&input(PIN\_B3)==1)//single running light

{

if(num==256){num=1;}//edge judge part

if(num==0.5){num=128;}

output\_c((int)num);

int dir=input(PIN\_B0);//direction control

if(dir==1){num=num\*2;}

if(dir==0){num=num/2;}

if(input(PIN\_B1)==1){sp=1;}//speed control

if(input(PIN\_B1)==0){sp=5;}

delay\_ms(250/sp);

if(!((input(PIN\_B4)==1)&&(input(PIN\_B5)==1)&&(input(PIN\_B6)==1)&&(input(PIN\_B7)==0)))

{

return;

}

}

//edge judge part

if(num==256){num=1;}

if(num==0.5){num=128;}

if((num==1)||(num==2)||(num==4)||(num==8)||(num==16)||(num==32)||(num==64)){num=num\*3;}

if(num==128){num=129;}

if((num==3)||(num==6)||(num==12)||(num==24)||(num==48)||(num==96)||(num==192)||(num==384)||(num==1.5)||(num==258)||(num==64.5)){num=num;}

while(input(PIN\_B2)==0&&input(PIN\_B3)==0)//double running light

{

if(num==384){num=129;}//edge judge part

if(num==258){num=3;}

if(num==1.5){num=129;}

if(num==64.5){num=192;}

output\_c((int)num);

int dir=input(PIN\_B0);//direction control

if(dir==1){num=num\*2;}

if(dir==0){num=num/2;}

if(input(PIN\_B1)==1){sp=1;}//speed control

if(input(PIN\_B1)==0){sp=5;}

delay\_ms(250/sp);

if(!((input(PIN\_B4)==1)&&(input(PIN\_B5)==1)&&(input(PIN\_B6)==1)&&(input(PIN\_B7)==0)))

{

return;

}

}

if(!((input(PIN\_B4)==1)&&(input(PIN\_B5)==1)&&(input(PIN\_B6)==1)&&(input(PIN\_B7)==0)))

{

output\_c(0);

return;

}

}

}

void s15()//inputB4B5B6B7,B7=LSB

{

while(TRUE)

{

if((input(PIN\_B4)==0)&&(input(PIN\_B5)==0)&&(input(PIN\_B6)==0)&&(input(PIN\_B7)==0))//if input=0000,call S0

{

s0();

}

if((input(PIN\_B4)==0)&&(input(PIN\_B5)==0)&&(input(PIN\_B6)==0)&&(input(PIN\_B7)==1))//if input=0001,call S1

{

s1();

}

if((input(PIN\_B4)==0)&&(input(PIN\_B5)==0)&&(input(PIN\_B6)==1)&&(input(PIN\_B7)==0))//if input=0010,call S2

{

s2();

}

if((input(PIN\_B4)==0)&&(input(PIN\_B5)==0)&&(input(PIN\_B6)==1)&&(input(PIN\_B7)==1))//if input=0011,call S3

{

s3();

}

if((input(PIN\_B4)==0)&&(input(PIN\_B5)==1)&&(input(PIN\_B6)==0)&&(input(PIN\_B7)==0))//if input=0100,call S4

{

s4();

}

if((input(PIN\_B4)==0)&&(input(PIN\_B5)==1)&&(input(PIN\_B6)==0)&&(input(PIN\_B7)==1))//if input=0101,call S5

{

s5();

}

if((input(PIN\_B4)==0)&&(input(PIN\_B5)==1)&&(input(PIN\_B6)==1)&&(input(PIN\_B7)==0))//if input=0110,call S6

{

s6();

}

if((input(PIN\_B4)==0)&&(input(PIN\_B5)==1)&&(input(PIN\_B6)==1)&&(input(PIN\_B7)==1))//if input=0111,call S7

{

s7();

}

if((input(PIN\_B4)==1)&&(input(PIN\_B5)==0)&&(input(PIN\_B6)==0)&&(input(PIN\_B7)==0))//if input=1000,call S8

{

s8();

}

if((input(PIN\_B4)==1)&&(input(PIN\_B5)==0)&&(input(PIN\_B6)==0)&&(input(PIN\_B7)==1))//if input=1001,call S9

{

s9();

}

if((input(PIN\_B4)==1)&&(input(PIN\_B5)==0)&&(input(PIN\_B6)==1)&&(input(PIN\_B7)==0))////if input=1001,call S10

{

s10();

}

if((input(PIN\_B4)==1)&&(input(PIN\_B5)==0)&&(input(PIN\_B6)==1)&&(input(PIN\_B7)==1))////if input=1011,call S11

{

s11();

}

if((input(PIN\_B4)==1)&&(input(PIN\_B5)==1)&&(input(PIN\_B6)==0)&&(input(PIN\_B7)==0))//if input=1100,call S12

{

s12();

}

if((input(PIN\_B4)==1)&&(input(PIN\_B5)==1)&&(input(PIN\_B6)==0)&&(input(PIN\_B7)==1))//if input=1101,call S13

{

s13();

}

if((input(PIN\_B4)==1)&&(input(PIN\_B5)==1)&&(input(PIN\_B6)==1)&&(input(PIN\_B7)==0))//if input=1110,call S14

{

s14();

}

if((input(PIN\_B4)==1)&&(input(PIN\_B5)==1)&&(input(PIN\_B6)==1)&&(input(PIN\_B7)==1))////if input=1112,output=0

{

output\_c(0);

}

}

}

void main()

{

s15();

}