Group 44: Qin Xuan Xu (261053393), William Zhang (260975150)

# Lab 3: g44_lab3

In this lab, we created a digital system using Qsys that implements the ARCCOS circuit that was written in the previous labs. The output is mapped on the segment displays of the De1 Soc board.

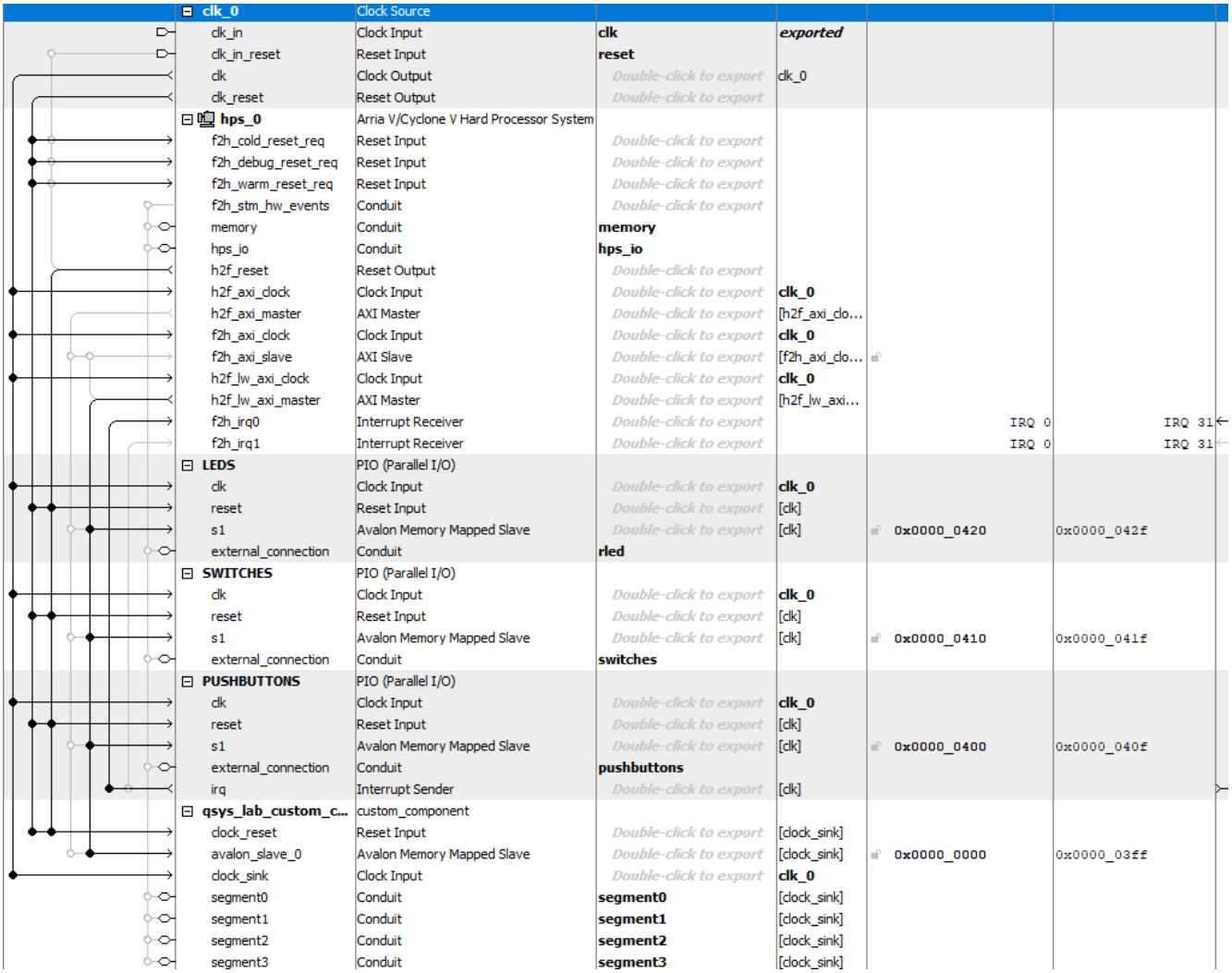


Figure 1: Qsys block diagram

Group 44: Qin Xuan Xu (261053393), William Zhang (260975150)

The following 4 pictures show the results of 4 tests done on the system.

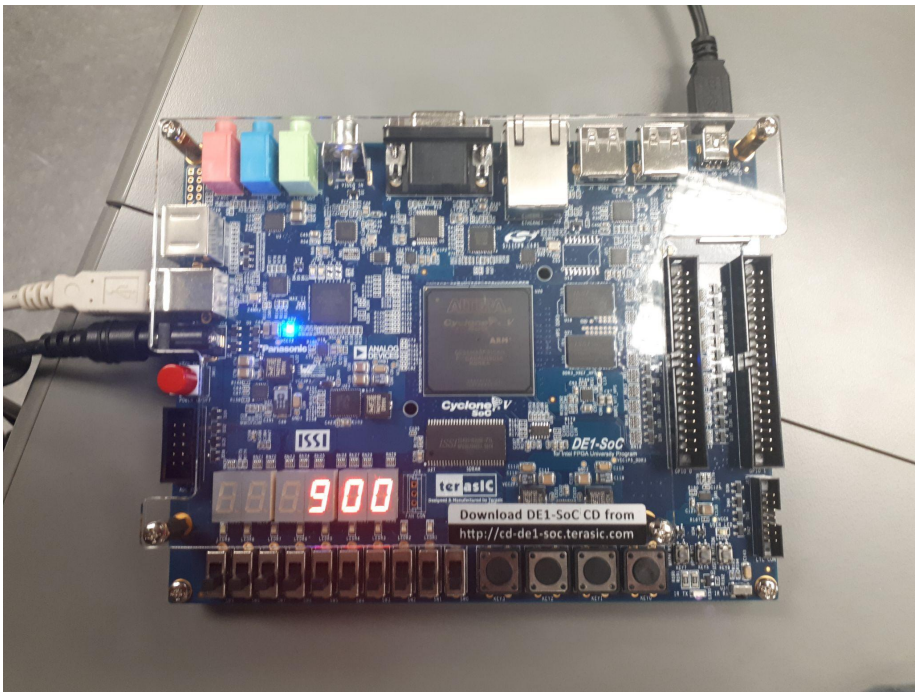


Figure 2: Output with X = 0

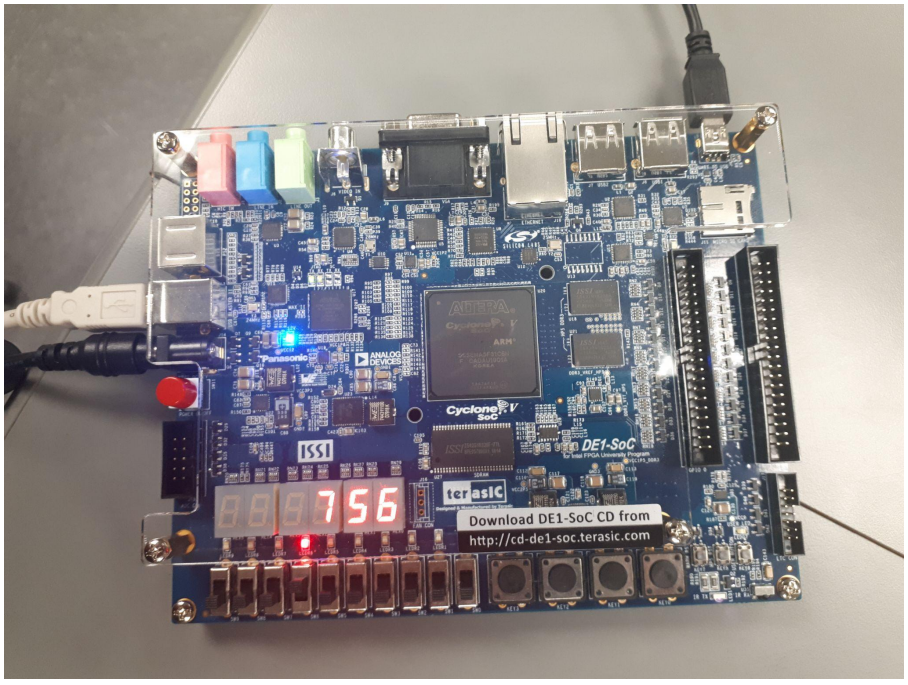


Figure 3: Output with X = 64

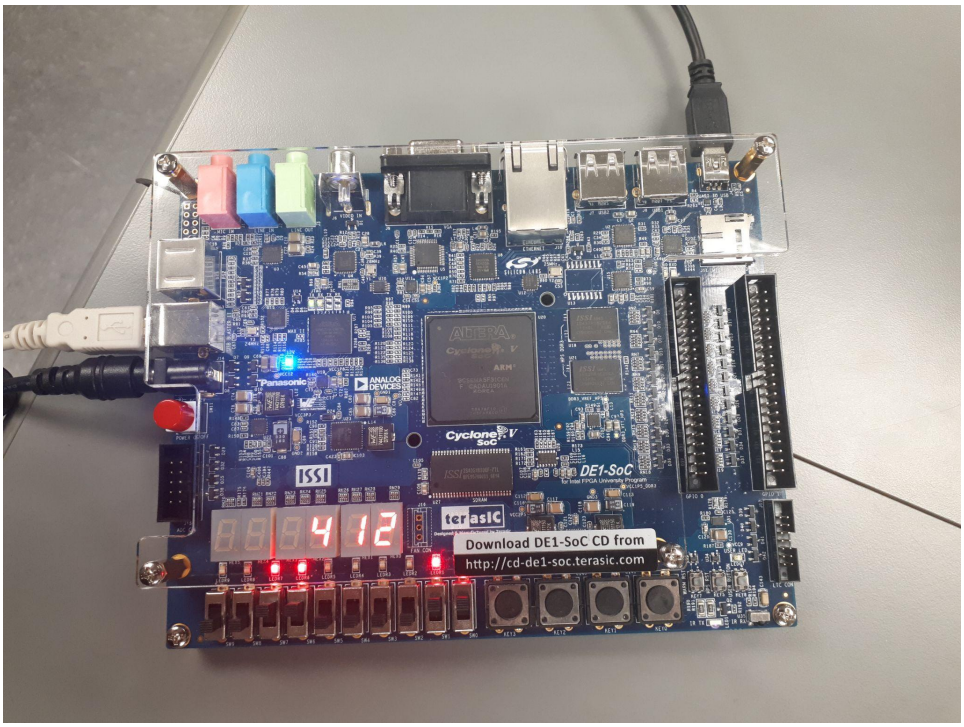


Figure 4: Output with X = 195

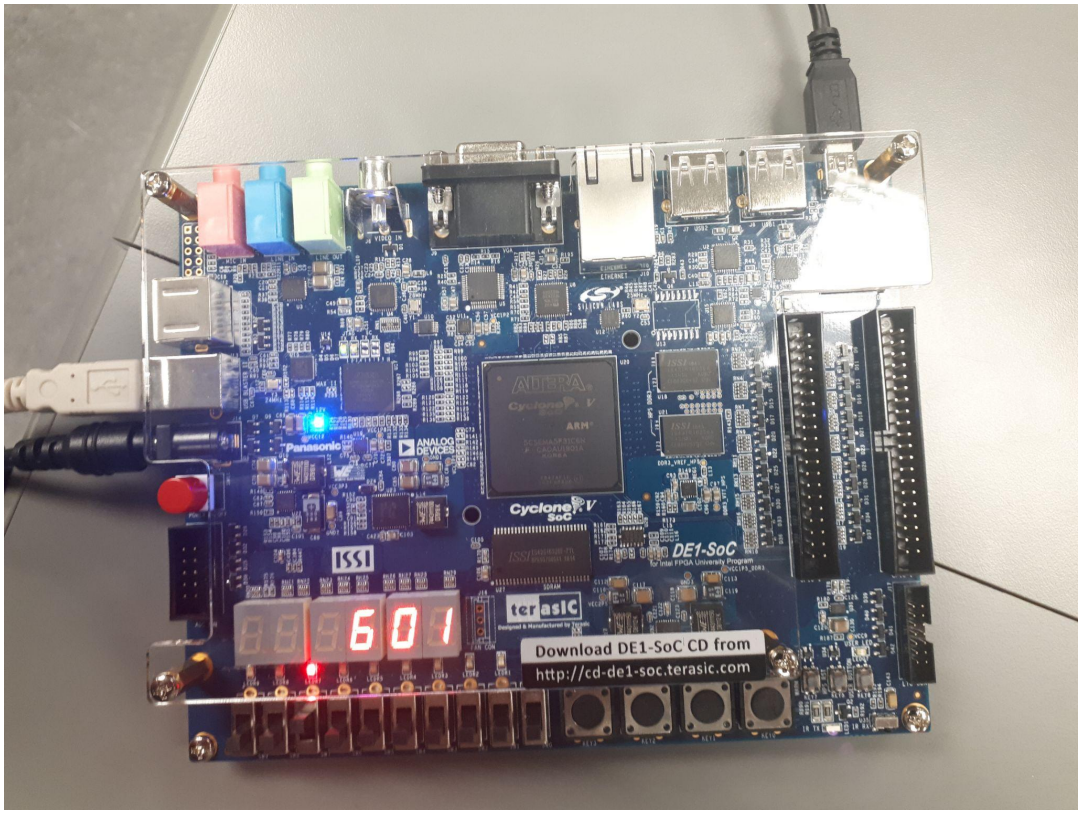


Figure 5: Output with X = 128

Group 44: Qin Xuan Xu (261053393), William Zhang (260975150)

The following figures show the C program used for testing the circuit.

```
#include<sys/time.h>
#include<stdio.h>

int SetTimer(struct timeval *tv, time_t sec)
{
    gettimeofday(tv,NULL);
    tv->tv_sec+=sec;

    return 1;
}
////////////////////////////////////////////////
int CheckTimer(struct timeval *tv, time_t sec)
{
    struct timeval ctv;
    gettimeofday(&ctv,NULL);

    if( (ctv.tv_sec >= tv->tv_sec) )
    {
        gettimeofday(tv,NULL);
        tv->tv_sec+=sec;
        return 1;
    }
    else {
    return 0;
   }
}

////////////////////////////////////
int pack_hex(int hex3, int hex2, int hex1, int hex0)
{
   return ((hex3 << 24) | (hex2 << 16) | (hex1 << 8) | hex0);
}
////////////////////////////////////
int hex_to_7segment(int hex)
{
   switch(hex) {
   case 0 :
      return 0b11000000;
   case 1 :
      return 0b11111001;
   case 2 :
      return 0b10100100;
   case 3 :
      return 0b10110000;
```

```
    case 3 :
       return 0b10110000;
    case 4 :
       return 0b10011001;
    case 5 :
       return 0b10010010;
    case 6 :
       return 0b10000010;
    case 7 :
       return 0b11111000;
    case 8 :
       return 0b10000000;
    case 9 :
       return 0b10010000;
    default :
       return 0b11111111;
}
return(0);
}
//////////////////////////////////////////

// Simple test program
int main(void)
{
    volatile int * component_op1 = (int *) 0xFF200000; // component base address
    volatile int * component_op2 = (int *) 0xFF200004; // component base address + 4
    volatile int * led = (int *) 0xFF200420; // red LED address
    volatile int * switchptr = (int *) 0xFF200410; // SW slider switch address
    volatile int * pushbuttons = (int *) 0xFF200400; // pushbuttons address
    int switch_value;
    int pb_val;
    int component_value;

    struct timeval tv;
    SetTimer(&tv,1);

    int secs = 0;
    int secs_tens = 0;
    int mins = 0;
    int mins_tens = 0;
    int hours = 0;
    int hours_tens = 0;

    while (1)
```

```
    struct timeval tv;
    SetTimer(&tv,1);

    int secs = 0;
    int secs_tens = 0;
    int mins = 0;
    int mins_tens = 0;
    int hours = 0;
    int hours_tens = 0;

    while (1)
    {
       if (CheckTimer(&tv,1)==1) {
          secs++;
          if (secs == 10) {
          secs = 0;
          secs_tens++;
             if (secs_tens == 6) {
                secs_tens = 0;
                mins++;
                if (mins == 10) {
                   mins = 0;
                   mins_tens++;
                   if (mins_tens == 6) {
                      mins_tens = 0;
                      hours++;
                      if (hours == 10 || ((hours_tens == 1) && (hours == 2))) {
                         hours = 0;
                         hours_tens++;
                         if (hours_tens == 2) {
                            hours_tens = 0;
                         }
                      }
                   }
                }
             }
          }
       }

       pb_val = *(pushbuttons); // read pushbutton values
       switch_value = *(switchptr); // read board switch values
       *(component_op1) = switch_value; // write op1 data to the component
       *(component_op2) = switch_value; // write op2 data to the component
       *(led) = switch_value;
    }
}
```

Figure 6: C program for circuit testing