16 December 2023

Welcome to

RISC-V - Hardware Design Program

Instructor -

Mayank Kabra

CHIPCRON PRIVATE LIMITED

Kunal ghosh

VLSI SYSTEM DESIGN

CHIPCRON PRIVATE LIMITED

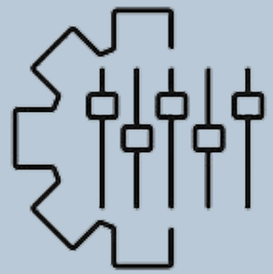
About Company

We at Chipcron aim to design a reconfigurable platform to customize a RISC-V SOC to create Application specific System on Chips.

Our Vision

Platform

Easy to reconfigure platform for RISC-V SOC's.



Customization

Application-specific customization control at the instruction level and Power, Frequency, Area.

Automation

Automatic setup to generate RTL to GDSII for faster tapeouts.

Course Overview

SOFTWARE

- Understanding RISC-V
- Overall design of application
- C code implementation
- Verification of C code using compilers

HARDWARE

- Designing SoC using ChipCron Tool
- Understanding and designing testbench
- Verification via simulation

Understanding RISC-V

Eg. Stop Watch app

Mozilla Firefox

Acrobat Reader DC

Oracle VM VirtualBox

Application Software or Apps

System Software

Windows 7

Linux

OS

C, C++, VB, Java

COMPILER

Instr 1
Instr 2

*.exe file

- Handle IO operations
- Allocate memory
- Low level system functions

```c
#include <stdio.h>
#include <time.h> //for sleep() function

int main()
{
    int hour, minute, second;
    hour=minute=second=0;
    while(1)
    {
        //clear output screen
        system("clear");
        //print time in HH : MM : SS format
        printf("%02d : %02d : %02d ",hour,minute,second);
         //clear output buffer in gcc
        fflush(stdout);
         //increase second
        second++;
        //update hour, minute and second
        if(second==60){
            minute+=1;
            second=0;
        }
        if(minute==60){
            hour+=1;
            minute=0;
        }
        if(hour==24){
            hour=0;
            minute=0;
            second=0;
        }
        sleep(1);    //wait
    }
    return 0;
}
```

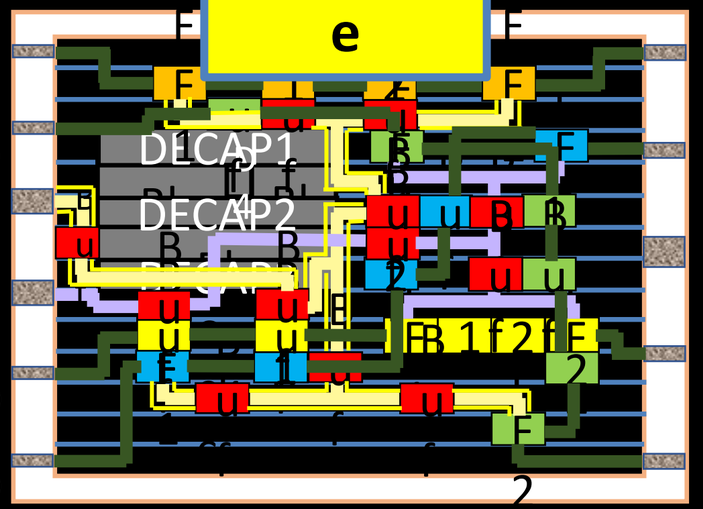
Compiler input, a 'c' function for stopwatch

ASSEMBLER

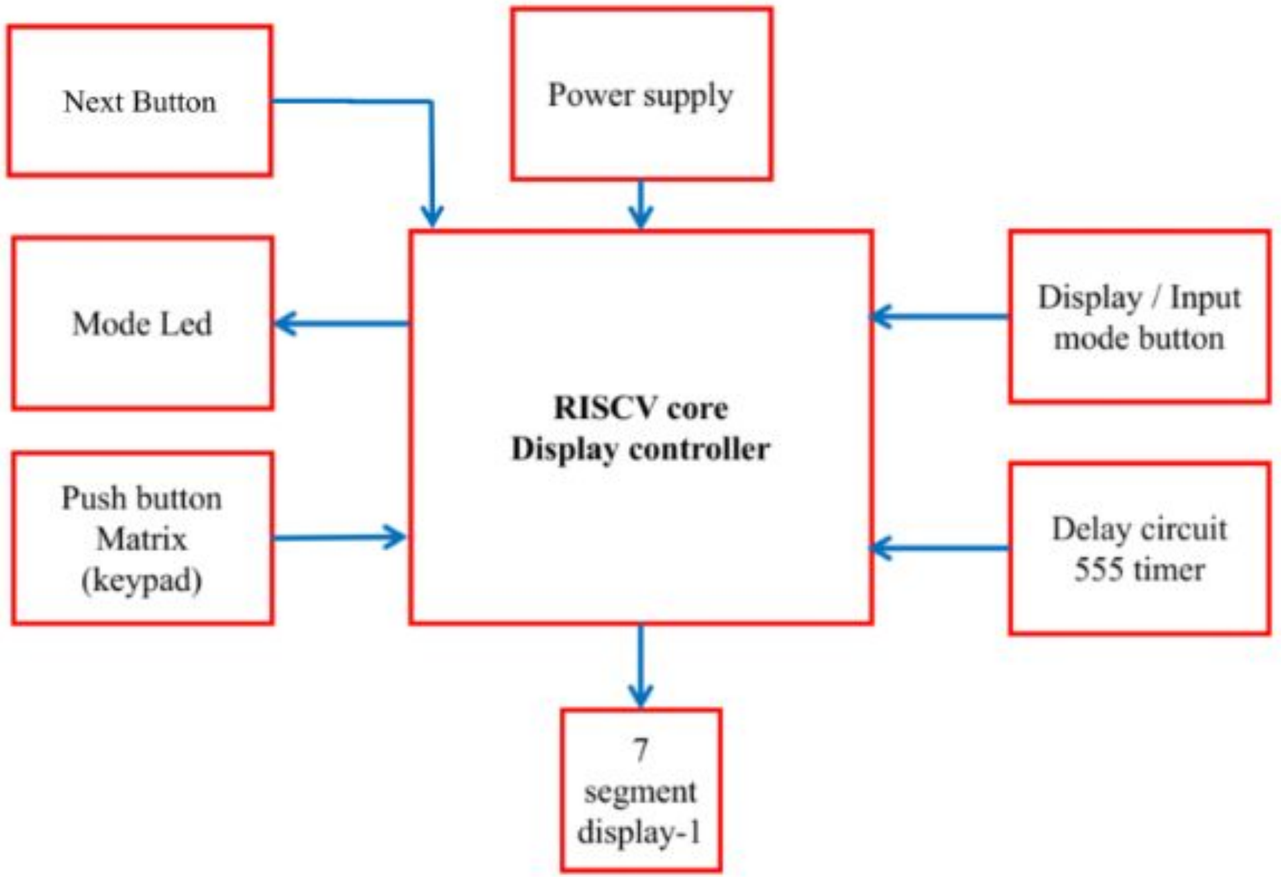
```
0000000000000001 <main>:
   1:   715d                addi    sp,sp,-80
   3:   e45e                sd      s7,8(sp)
   5:   00000bb7            lui     s7,0x0
   9:   000b8513            mv      a0,s7
   d:   e486                sd      ra,72(sp)
   f:   e0a2                sd      s0,64(sp)
  11:   fc26                sd      s1,56(sp)
  13:   f84a                sd      s2,48(sp)
  15:   f052                sd      s4,32(sp)
  17:   ec56                sd      s5,24(sp)
  19:   e85a                sd      s6,16(sp)
  1b:   e062                sd      s8,0(sp)
  1d:   f44e                sd      s3,40(sp)
  1f:   00000b37            lui     s6,0x0
  23:   00000097            auipc   ra,0x0
  27:   000080e7            jalr    ra
  2b:   4681                li      a3,0
  2d:   4601                li      a2,0
  2f:   4581                li      a1,0
  31:   000b0513            mv      a0,s6
  35:   00000097            auipc   ra,0x0
  39:   000080e7            jalr    ra
  3d:   00000ab7            lui     s5,0x0
```

Compiler and assembler output, A RISC-V assembly language program

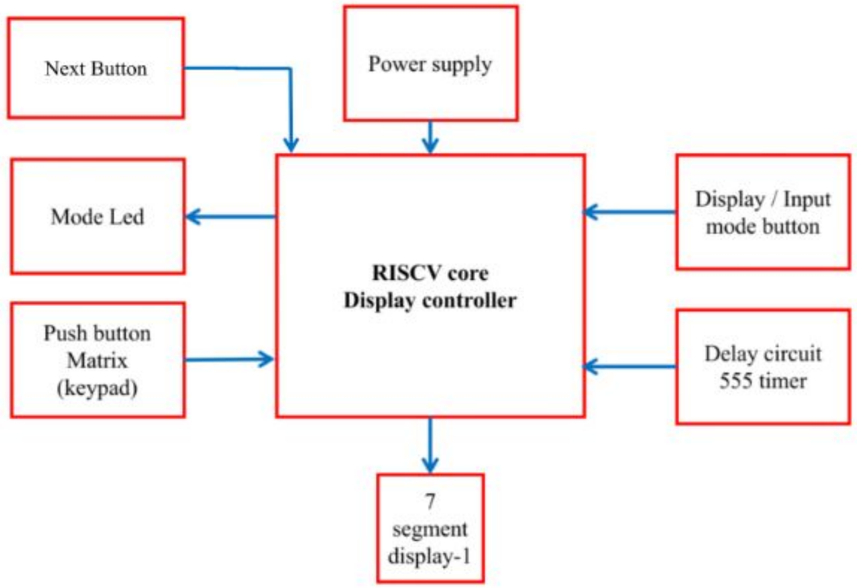
Hardware

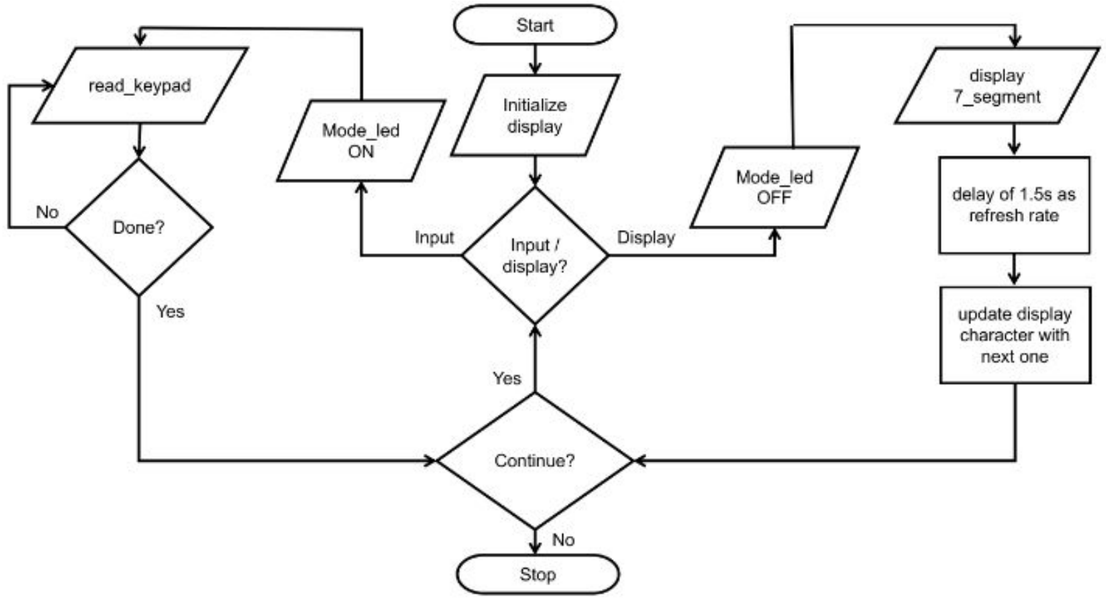


Overall design of application

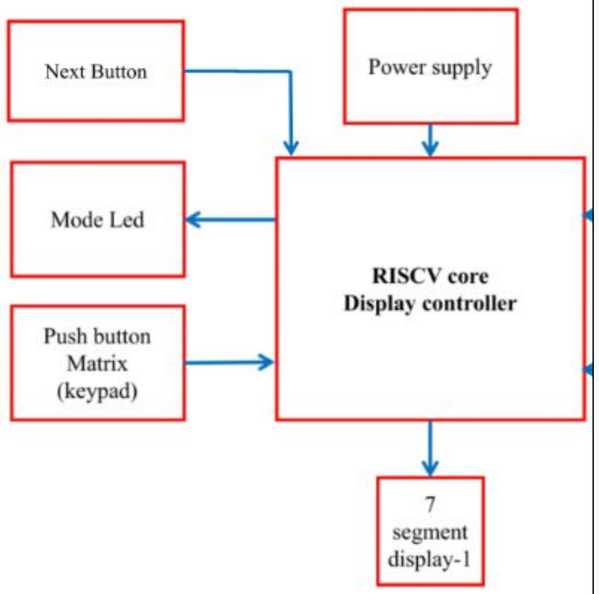


Overall design of application

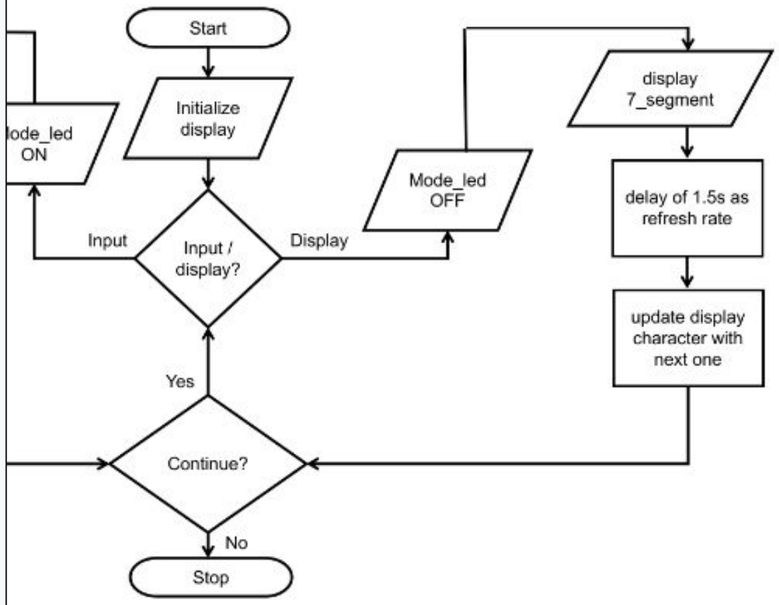




C Code of application

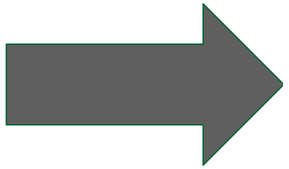


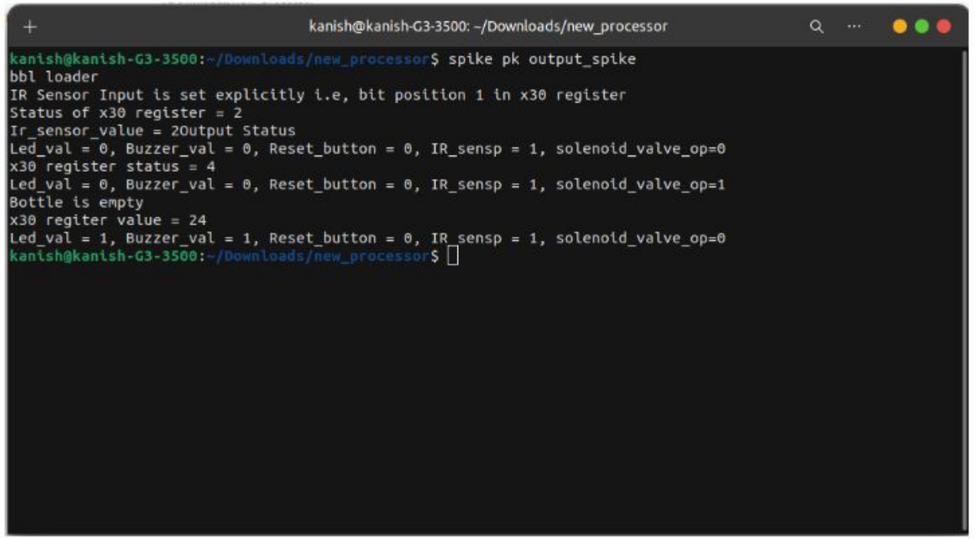

```
int main()
{
	int mode;
	int display1;
	int delay;
	int next;
	int keypad;
	int a=0,b=0,c=0,d=0,e=0,f=0,g=0,h=0,i=0,j=0;
	int count1=0;


	//initialize with hypen
	display1_output(1);


	while(1)
	{
		mode=read_mode();
		display_mode(mode);
		if(mode==1)//input new text
		{
			keypad=read_keypad();
			if(keypad!=0)
			{
				if(count1==0) a=keypad;
				else if(count1==1) b=keypad;
				else if(count1==2) c=keypad;
				else if(count1==3) d=keypad;
				else if(count1==4) e=keypad;
				else if(count1==5) f=keypad;
				else if(count1==6) g=keypad;
				else if(count1==7) h=keypad;
				else if(count1==8) i=keypad;
				else if(count1==9) j=keypad;
				else if(count1==10) count1=0;
				if(keypad!=1)
				{
					count1++;
					display1_output(keypad);
					next=read_next();
					while(next==0)
					{
						next=read_next();
					}
```

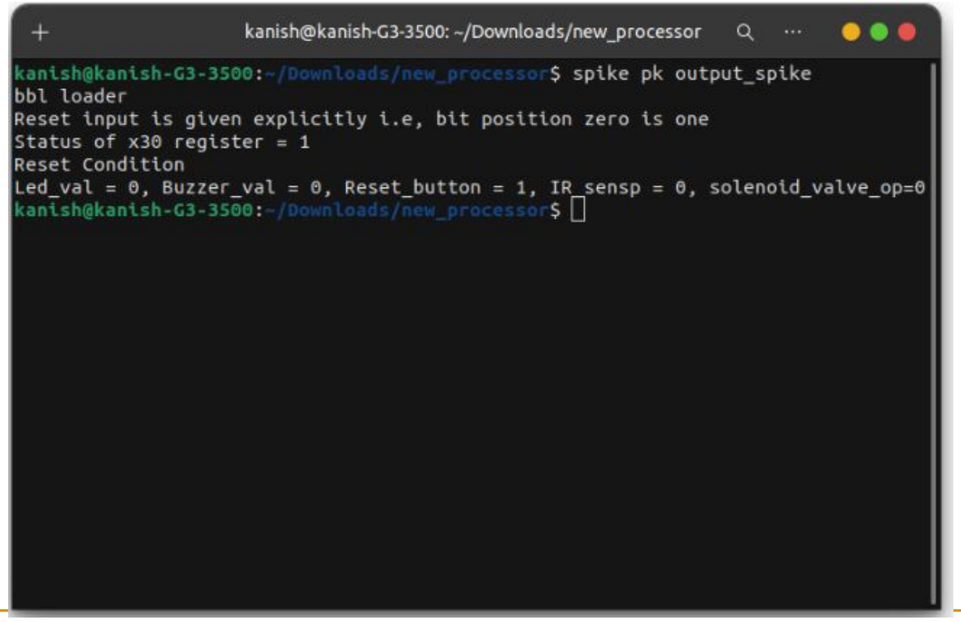


Verification of C code using compilers

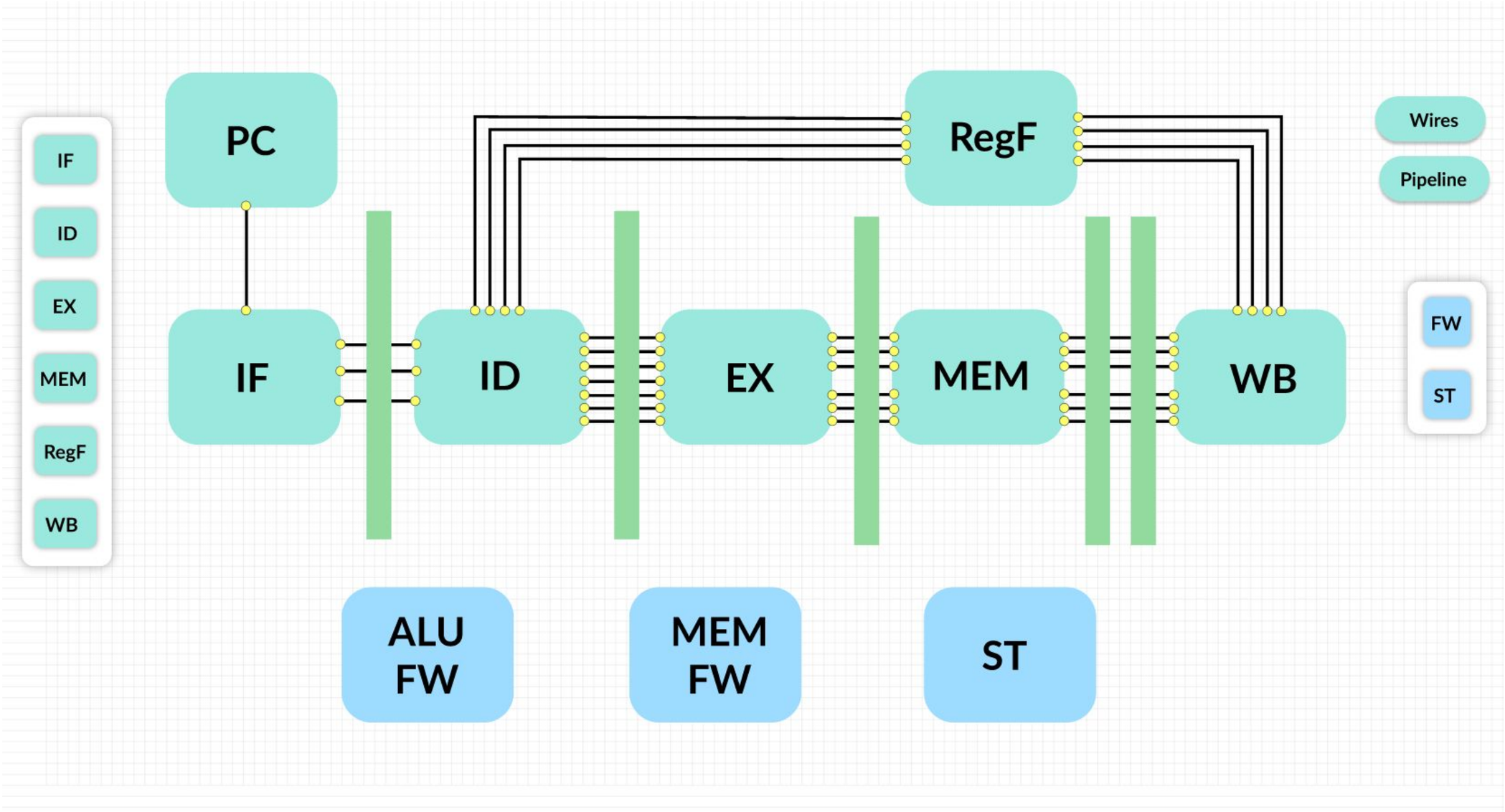
```c
int main()
{
	int mode;
	int display1;
	int delay;
	int next;
	int keypad;
	int a=0,b=0,c=0,d=0,e=0,f=0,g=0,h=0,i=0,j=0;
	int count1=0;


	//initialize with hypen
	display1_output(1);


	while(1)
	{
		mode=read_mode();
		display_mode(mode);
		if(mode==1)//input new text
		{
			keypad=read_keypad();
			if(keypad!=0)
			{
				if(count1==0) a=keypad;
				else if(count1==1) b=keypad;
				else if(count1==2) c=keypad;
				else if(count1==3) d=keypad;
				else if(count1==4) e=keypad;
				else if(count1==5) f=keypad;
				else if(count1==6) g=keypad;
				else if(count1==7) h=keypad;
				else if(count1==8) i=keypad;
				else if(count1==9) j=keypad;
				else if(count1==10) count1=0;
				if(keypad!=1)
				{
					count1++;
					display1_output(keypad);
					next=read_next();
					while(next==0)
					{
						next=read_next();
					}
```



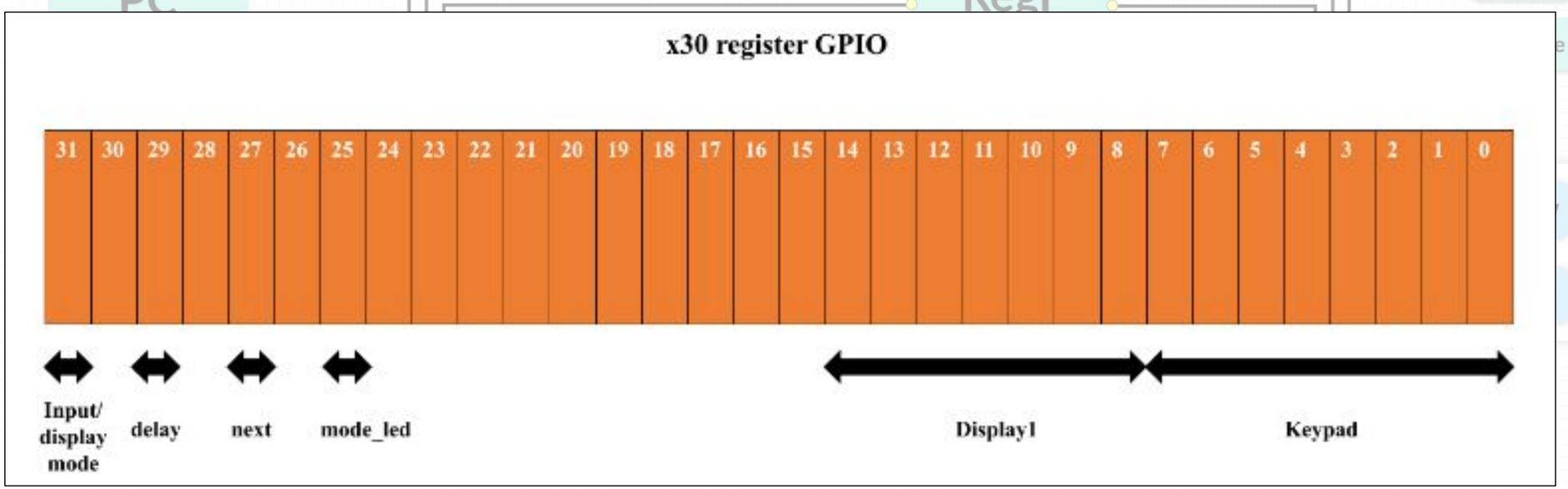




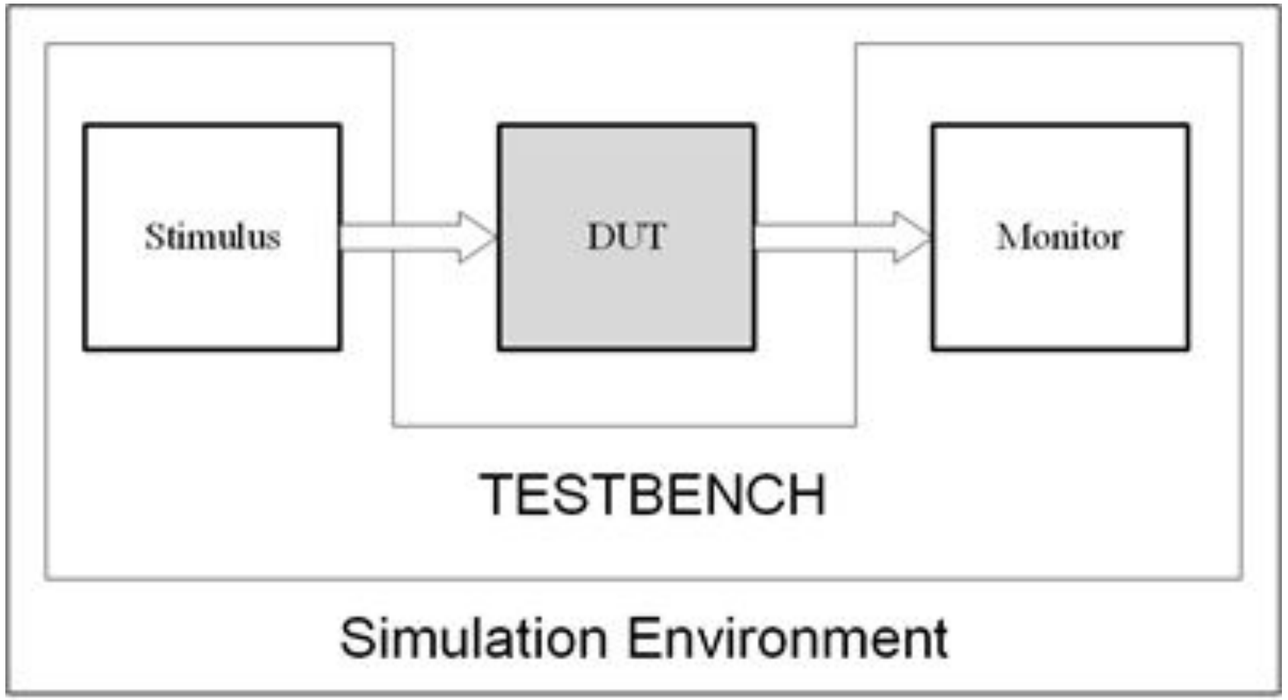
Designing SoC using ChipCron Tool



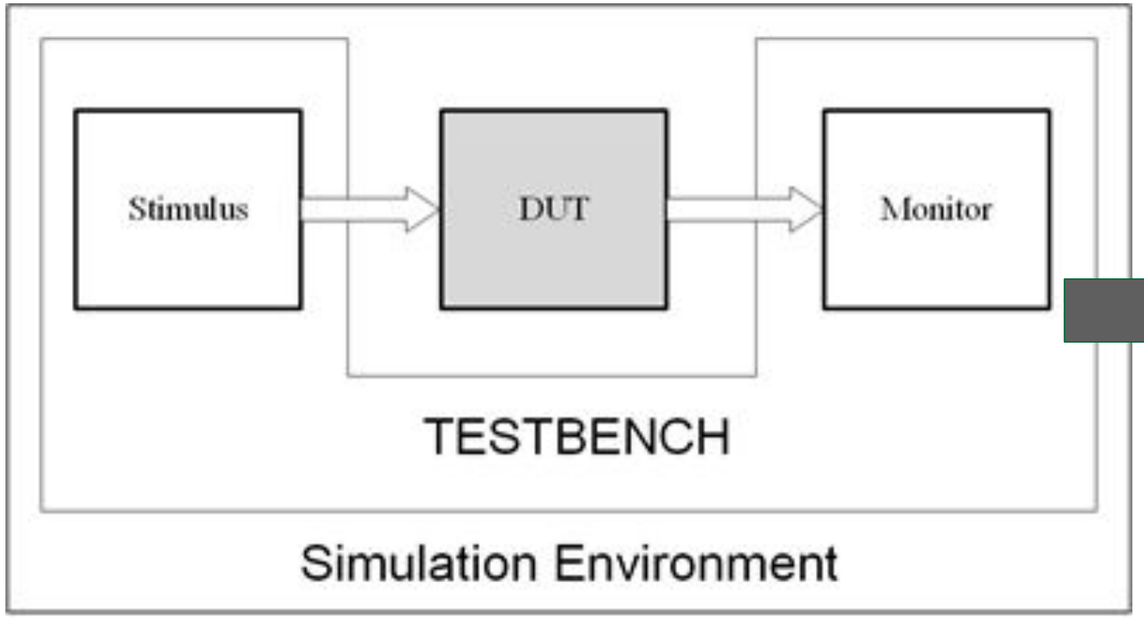
Designing SoC using ChipCron Tool

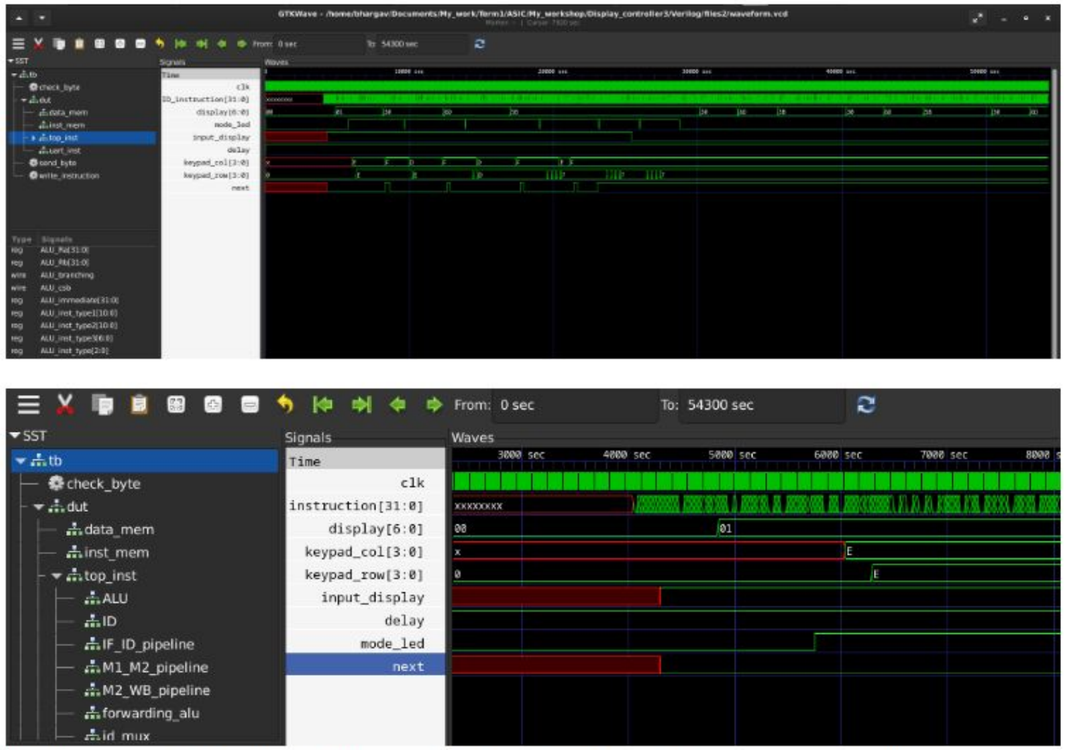


Verification via simulation



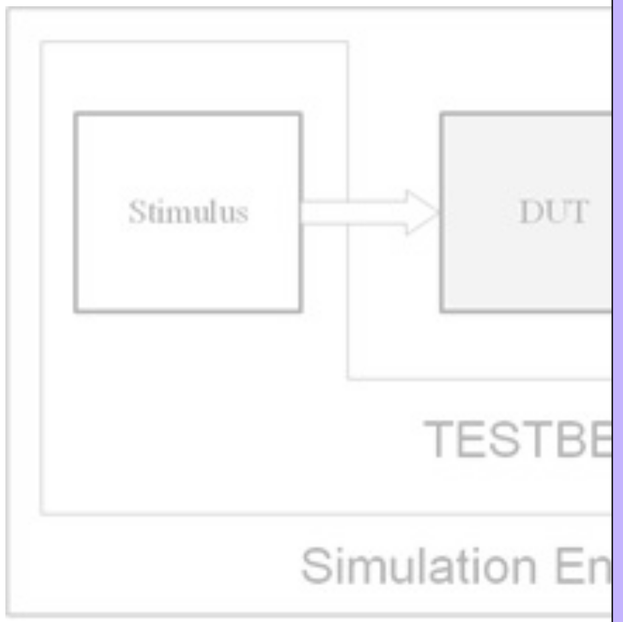
Verification via simulation





Initially, we keep input_display=1 for taking inputs from keypad. 7 segment displays hypen (hex code 0x01) as shown.

Verification via simulation



Testbench Verification

Pre Synthesis and Post synthesis

d. 7 segment displays hypen (hex code 0x01) as

More Discussion

- Timings for discussion - Wed and Sun - 10pm IST
- Whatsapp or Slack group ? Both
- Plan for 6-weeks (Details to be shared in next class)
- More opportunities and internships

Task / Assignment - 1

- Setup Github and create a github repo (Public).
- Install vdi file in Oracle virtualbox.
- Design c code on Godbolt - (a) Counter

  (b) Matrix-Multiplication
- Document all the steps.

Any Question ?