Faculty of Computer Science and Artificial Intelligence

Computer Architecture

Final project

Banha Bank queue Monitor (BBqM)

- ***Team member***

leader	عمرو عونى عبدالتواب	*Section 14*
member	عمروخالد محمود	*Section 14*
member	غاده احمد السيد	*Section 14*

The name of the supervisor of the article:

Dr Ahmed shalapy

Eng mayada

➢ System description

It is bank system their is people in the queue . one queue have from 0 to 7 person in it and the tellers for system their range from 1 to 3 it can't be empty

There's sensors the give signal when person get the queue number of people increased by one and when person get from the queue people decreased by one

There's function in our system that calculate time person take in the queue by equation = 3*(pcount+tcount-1)/ tcount

This system is to monitor the movement inside the bank to avoid congestion, and depends on the presence of counters for the number of employees and the number of people in line. There should be always tellers in front of queue range (from 1 to 3) .This bank system depends on several inputs, including the number of people in the queue and the number of employees, and depends on a main entrance to determine the operation that the system will perform(increasing ,decreasing, initialization(reset))

➢ System inputs

input	Meaning
Pcount	This is the second entry, and it expresses the number of people to be served, and it can contain numbers between 0 and 7
Tcount	This is the first input of the system and can have numbers between 1 to 3 It indicates the number of tellers available in system[it mustn't be empty]
menu	This is the third entry, which determines the operation that the system will perform by entering a specific number, and there are 7 operations that it performs, so a number can be entered between 1 to 7 to determine the operation

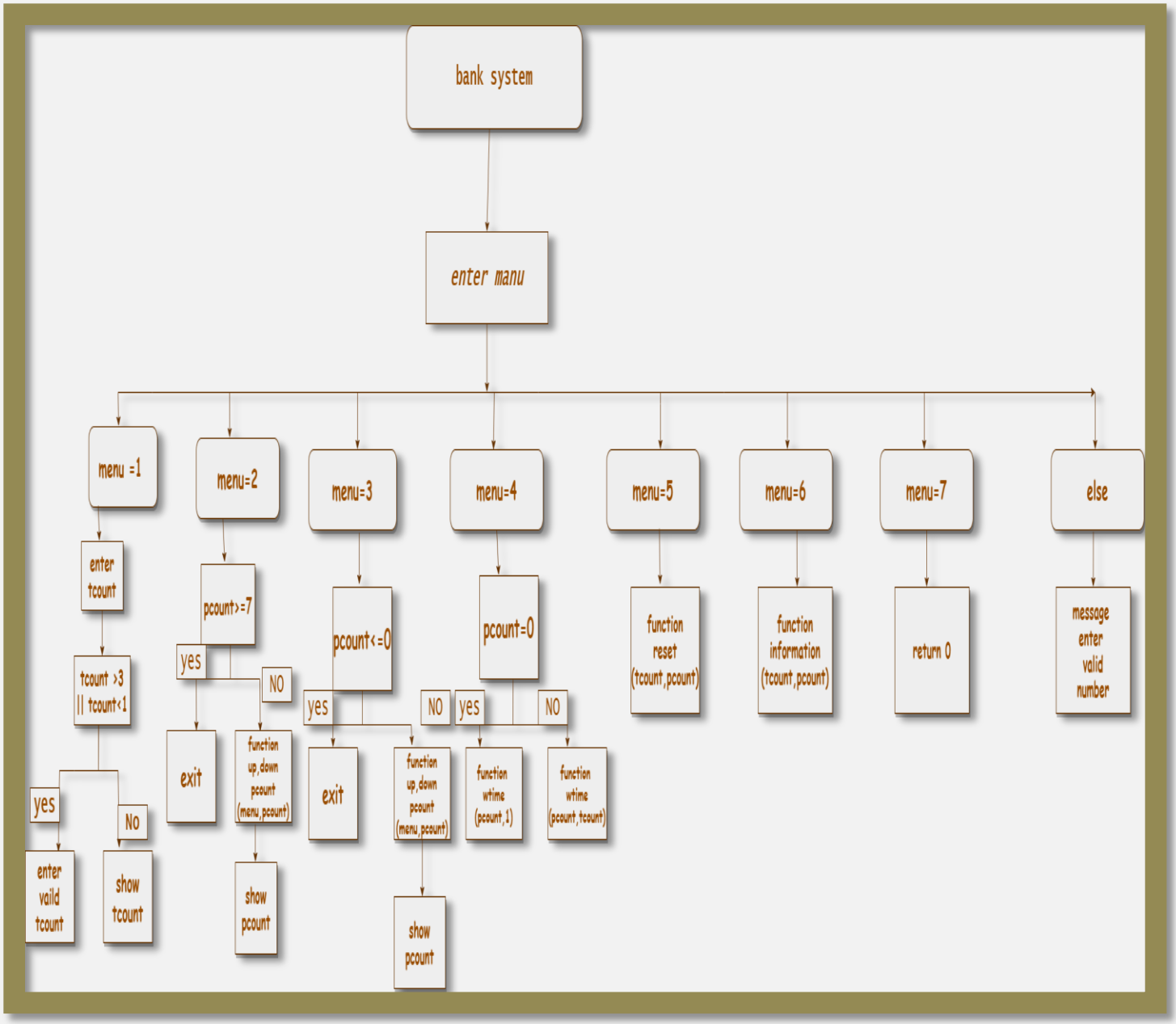
➢ System output

output	meaning
wtime	This output we calculate from function called wtime according to the equation Time= ***3*(pcount+tcount-1)/ tcount*** The meaning of it time that people taken in the queue There is conditions that mustn't be in the system like when Tcount [number of tellers]is equal to zero We can't divide under zero.

➢ System Tools :

- We Use C Programming Language To Write The Code Of The Bank System Program .
- We Use Assembly Language To Describe The Function (Wtime) That Show The Time To Make The Queue Empty.
- We Use Machine Language To Convert Assembly Code Of *Wtime Function*To Make The Mips Processor Understand It.

➢ System statement of work



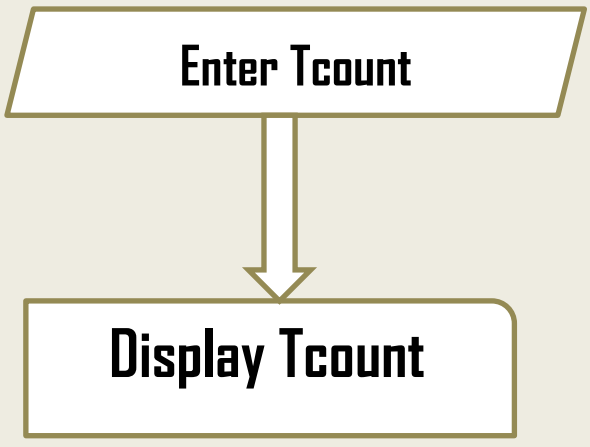
Description for each function

✓ Entering Tcount[number of tellers]

- Code in c

```
int number_tcount() {
    int tc = 1;
    cin >> tc;
    return tc;
}
```

-This Function will take number of tellers and display Tcount that enterd

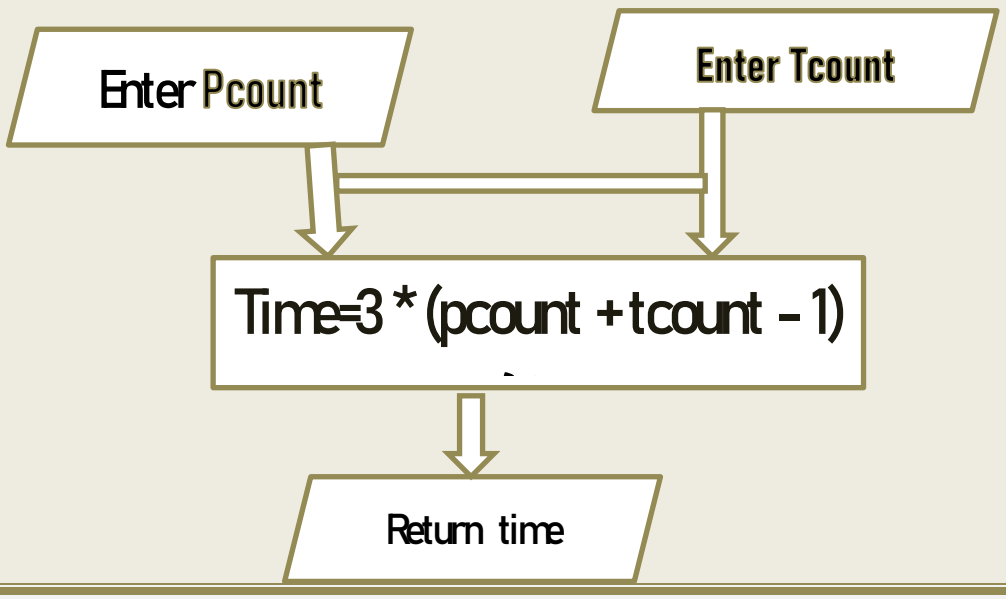


• Calculate Wtime

Code in c

```
int Wtime( int pc , int tc ) {
      int time = 3 * (pc + tc - 1) / tc;
      return time;
}
```

-This Function will take[number of tellers]Tcount and pcount [number of people in the queue] display wait time that taken



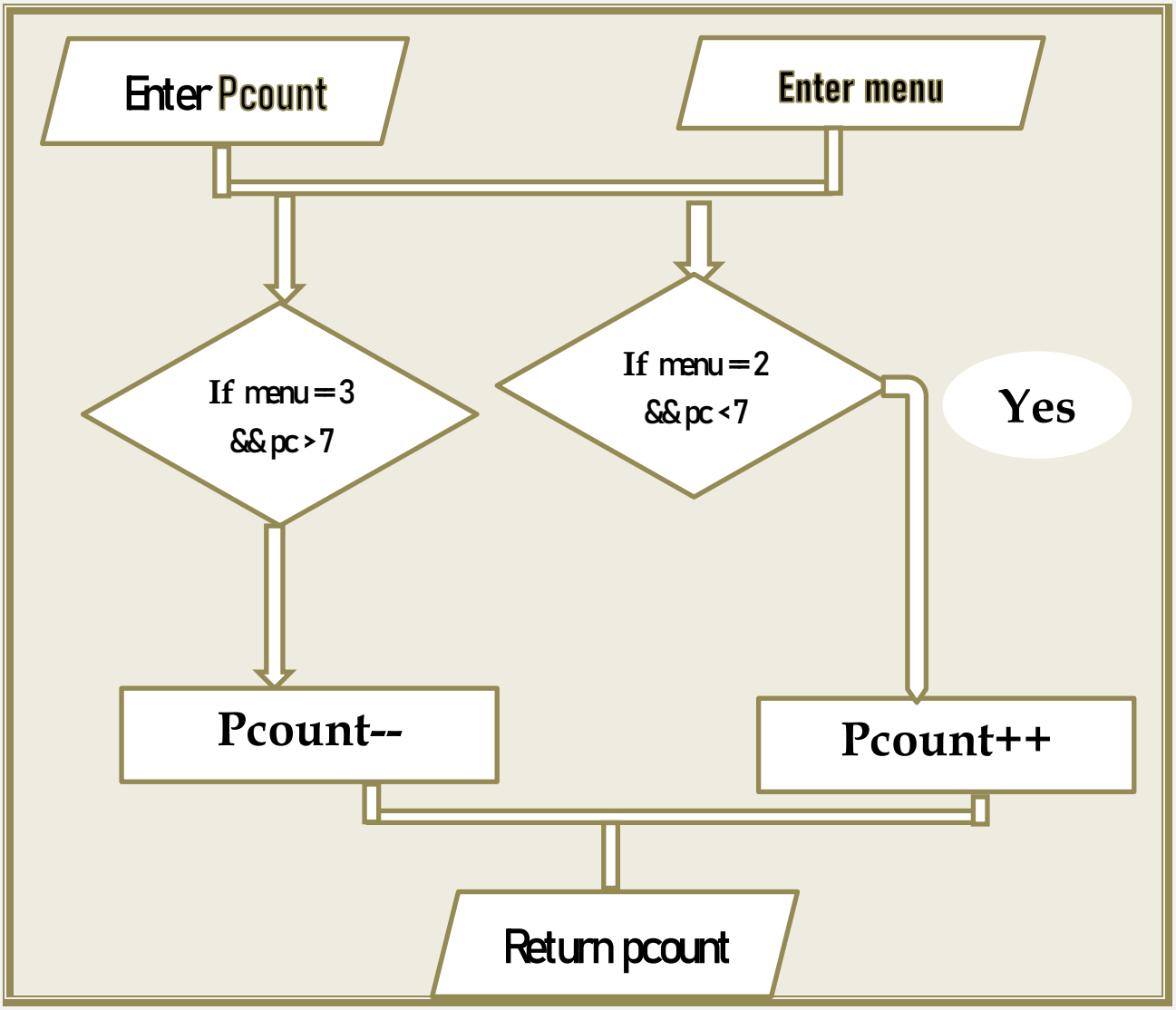
➢ Up_or_down pcount function

Code in c

```
int up_or_down_pcount(int pco) {
	int pc = pco;
	int key;
	while (key != '\n' && key != '\r') {
		key = getch();
		if (pc < 7 && key == 72) {
			pc++;
			cout << "the number of people in the queue is
: " << pc << endl;
		}
		else if (pc > 0 && key == 80) {
			pc--;
			cout << "the number of people in the queue is
: " << pc << endl;
		}
		else if (pc <= 0 && key == 80) {
			cout << "the queue is empty so you can't
decrease the number \n";
			continue;
		}
		else if (pc >= 7 && key == 72) {
			cout << "the queue is full so you can't
increase the number \n";
			continue;
		}

	}
	return pc;}
```

This function to decide if it is up or down counter to increase or decrease pcount[number of people in queue]this function take manu and call function then if pcount less than 7 it is increase case pcount by 1 else decrease pcount by 1 [ther is person get out of the queue] then return final value of pcount and can manipulate value of pcount by up & Down arrow



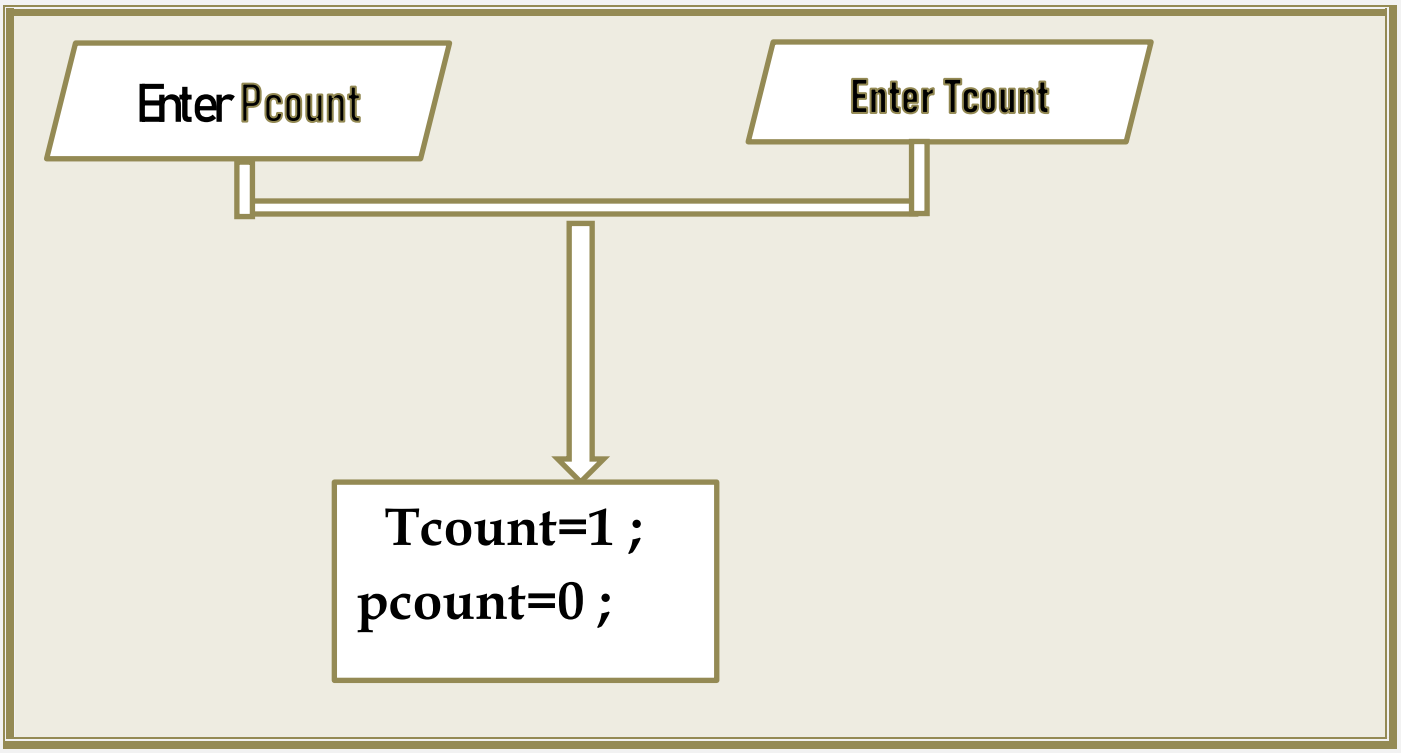
Function reset

```
void reset(int& tcount, int& pcount) {
      tcount =1;
      pcount =0;


  }
```

Code in c

This function to reset value of pcount[number of people in queue] andTcount[number of tellers]. That Tcount shouldn't be empty and pcount can be empty when their is no people in the queue.

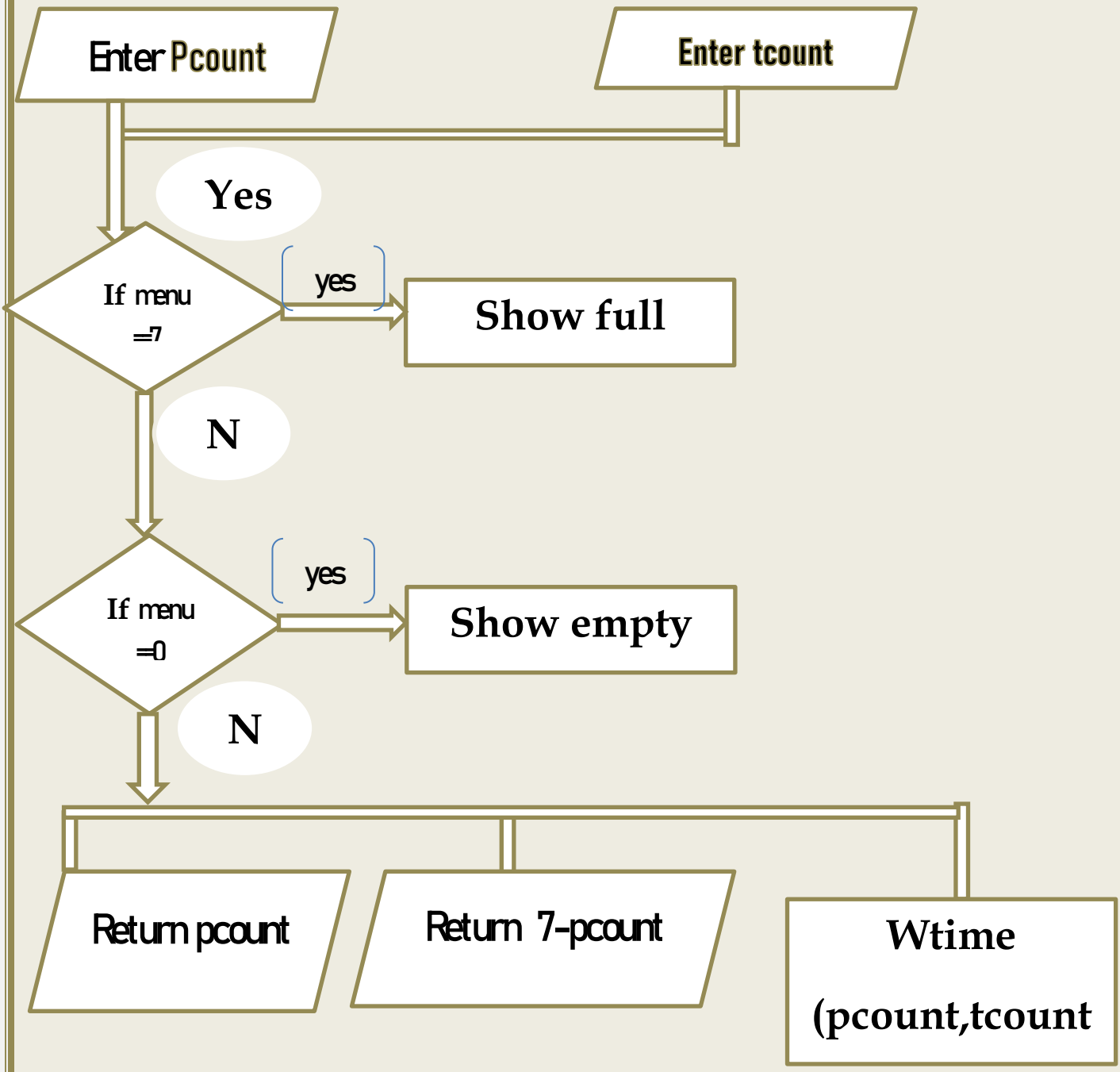


• Function information

Code in c

```
void information(int tcount, int pcount) {
        if (pcount == 7) {
                cout << "the size of the queue in the
bank is full \n";
        }
        else if (pcount == 0) {
                cout << "the size of the queue in the bank is empty \n";
        }
        else
                cout << "the number of the people in the queue is " << pcount << " and the the
number that can enter the bank at this moment is : " << 7 - pcount << "\n";
        cout << "the number of tellers is : " << tcount << "\n";
        cout << "the time to make the queue empty is : " << Wtime(pcount, tcount) << "\n";
}
```

- This function take pcount and tcount . if the queue is full[pcount=7] show message is full . if the queue is empty [pcount=0] show message is empty if all cases isn't true display pcount value , tcount ,and call function wtime and pass to it parameters pcount and tcount.


Enter Pcount
Enter tcount
Yes
If menu =7
yes
Show full
N
If menu =0
yes
Show empty
N
Return pcount
Return 7-pcount
Wtime (pcount,tcount

Faculty of Computer Science and Artificial Intelligence

Computer Architecture

➢ Main function

This function when user enter manu to decide if menu equal 1 enter tcount [number of tellers and if tcount >3 or tcount<1 then show message to enter valid number of tellers else ask for enter tcount by calling function tcount. Case menu equl 2 then tc if pcount>=7 then show message to enter valid number of people .else call function up down pcount pass menu and pcount and show value of pcount. Case menu equal 3 if pcount <=0 exit else call function up doqn pcount pass menu and pcount.and show pcount value. Case menu equal if pcount is then call function wtime and pass pcount and 1 teller , else pass real pcount and tcount. Case menu equal 5 call function reset and pass tcount and pcount . case menu equal 6 call

function information and pass tcount and pcount case 7 return 0 to user else show message enter valid menu

Code in c

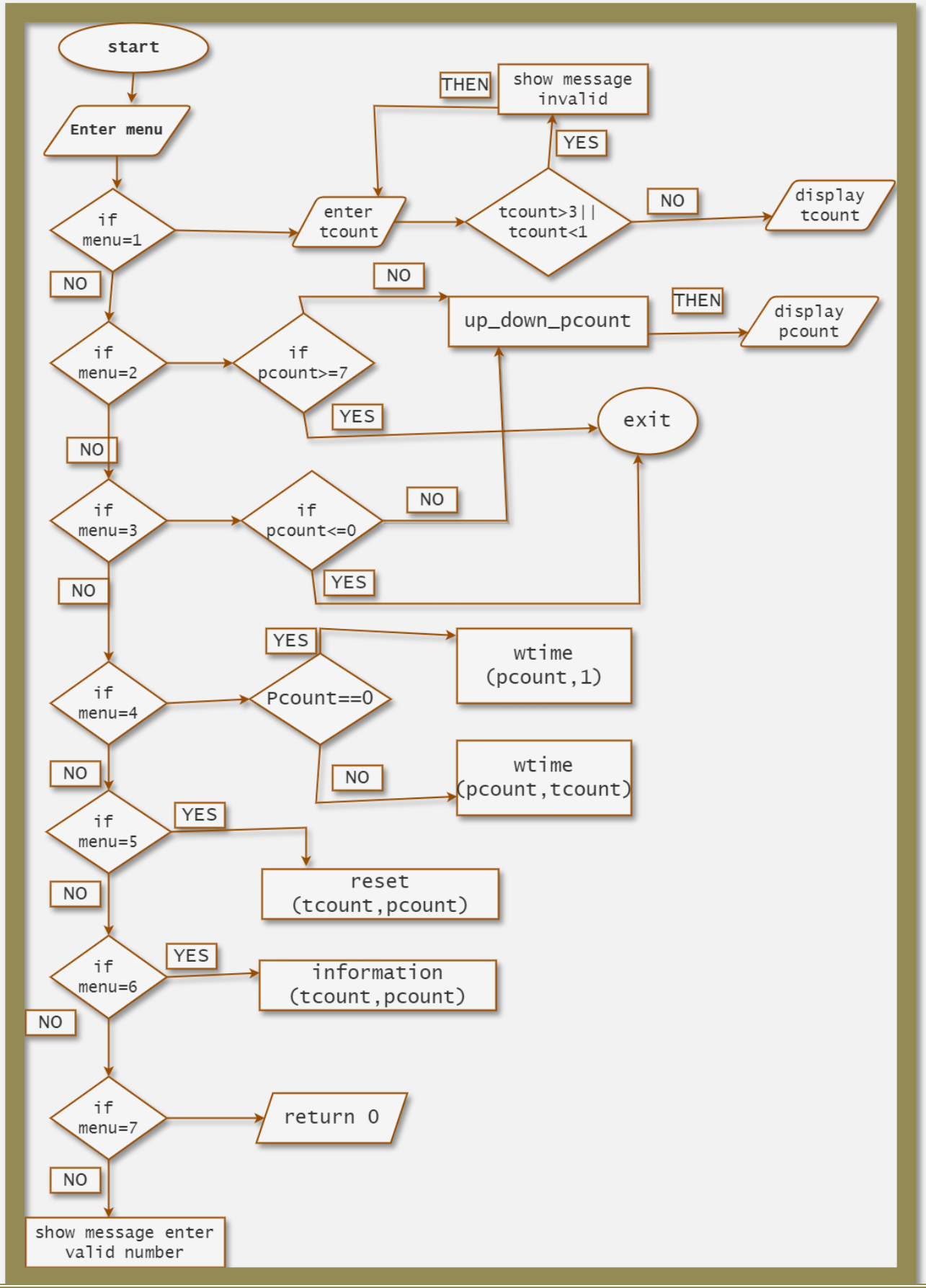
```
int main()
{
	int menu = 0;
	int pcount = 0, tcount = 1;
	while (1)
	{
		cout << "Enter number to do the action which refered to  \n 1:to enter the number of tellers \n 2:to up the number of people in the queue \n 3:to down the number of people in the queue \n 4:to show the waiting time in Seconds \n 5:to reset the system \n 6:to show information about the system \n 7:to exit \n";
		cin >> menu;
		if (menu == 1)
		{
			cout << "enter the number between 1:3 \n";
			tcount = number_tcount();
			while (tcount > 3 || tcount < 1) {
				cout << "this is unvalid number please enter valid number from 1 to 3 \n ";
				tcount = number_tcount();
			}
			cout << "the number of tcount is : " << tcount << endl;
		}
		else if (menu == 2) {
```

```
                if (pcount >= 7) {
                        cout << "the number of people is already 7 at the full
capcity \n";
                        continue;
                }
                pcount = up_or_down_pcount(menu, pcount);
                cout << "the number of people in the queue is : " << pcount <<
endl;
        }
        else if (menu == 3) {
                if (pcount <= 0) {
                        cout << "the number of people is already 0 \n";
                        continue;
                }
                pcount = up_or_down_pcount(menu, pcount);
                cout << "the number of people in the queue is : " << pcount <<
endl;
        }

        else if (menu == 4) {
                if (pcount == 0) {
                        cout << "the time to make the queue empty is : " <<
Wtime(pcount, 1) << " secounde because the bank is empty \n";
                }
                else
                {
```

```
                        cout << "the time to make the queue empty is : " <<
Wtime(pcount, tcount) << " secounde \n";
                }
        }
        else if (menu == 5) {
                reset(tcount, pcount);
                cout << "the system is reset successfully \n";
        }
        else if (menu == 6) {
                information(tcount, pcount);
        }
        else if (menu == 7) {
                return 0;
        }
        else {
                cout << "enter valid number \n";
        }

    }
    return 0;
}
```


start
Enter menu
if menu=1
enter tcount
tcount>3|| tcount<1
THEN
show message invalid
YES
NO
display tcount
NO
if menu=2
if pcount>=7
NO
up_down_pcount
THEN
display pcount
YES
exit
NO
if menu=3
if pcount<=0
NO
YES
NO
YES
if menu=4
Pcount==0
wtime (pcount,1)
NO
wtime (pcount,tcount)
if menu=5
YES
reset (tcount,pcount)
NO
if menu=6
YES
information (tcount,pcount)
NO
if menu=7
return 0
NO
show message enter valid number

C code	Assemply code

int Wtime(int pc , int tc)
{
int time = 3 * (pc + tc - 1)
/ tc;
return time;
}
	#Pcount =s0 Tcount=s1 x=s0
	addi $s0,$zero,7
	addi $s1,$zero,1
	addi $s2,$zero,0
	addi $s3,$zero,3
	add $s2,$s0,$s1
	addi $s2,$s2,-1
	mul $s2,$s2,$s3
	div $s2,$s2,$s1

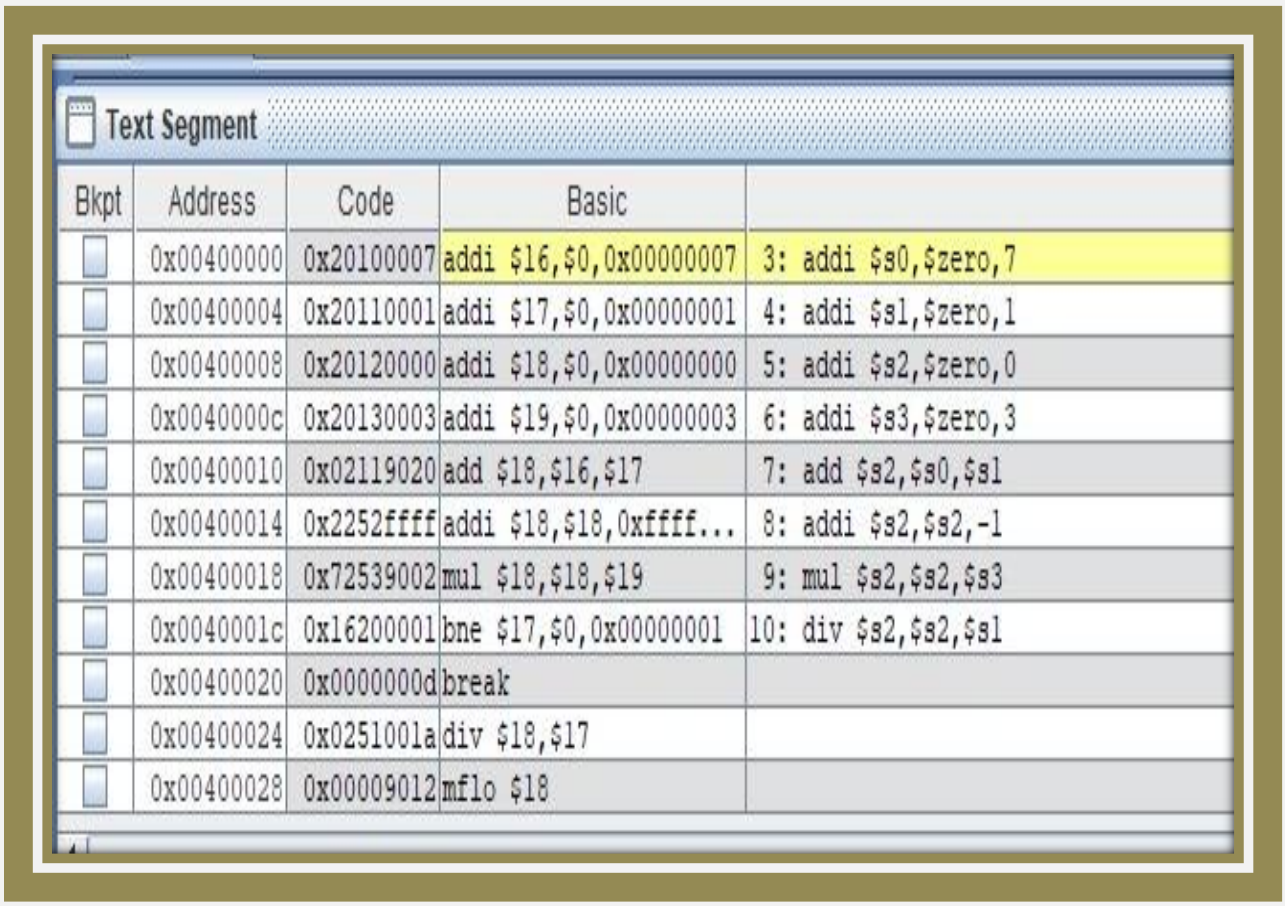
Machine code

```
20100005
20110002
20120000
20130003
02119020
2252FFFF
02539018
0251901a
```

Machine code

```
[0010.00][00.000][1.0000.][0000.0000.0000.0111]    0x20100007
[0010.00][00.000][1.0001.][0000.0000.0000.0001]    0x20110001
[0010.00][00.000][1.0010.][0000.0000.0000.0000]    0x20120000
[0010.00][00.000][1.0011.][0000.0000.0000.0011]    0x20130003
[0000.00][10.000][1.0001.][1001.0][000.00][10.0000]0x02119020
[0010.00][10.010][1.0010.][1111.1111.1111.1111]    0x2252FFFF
[0000.00][10.010][1.0011.][1001.0][000.00][01.1000]0x02539018
[0000.00][10.010][1.0001.][1001.0][000.00][01.1010]0x0251901A
```

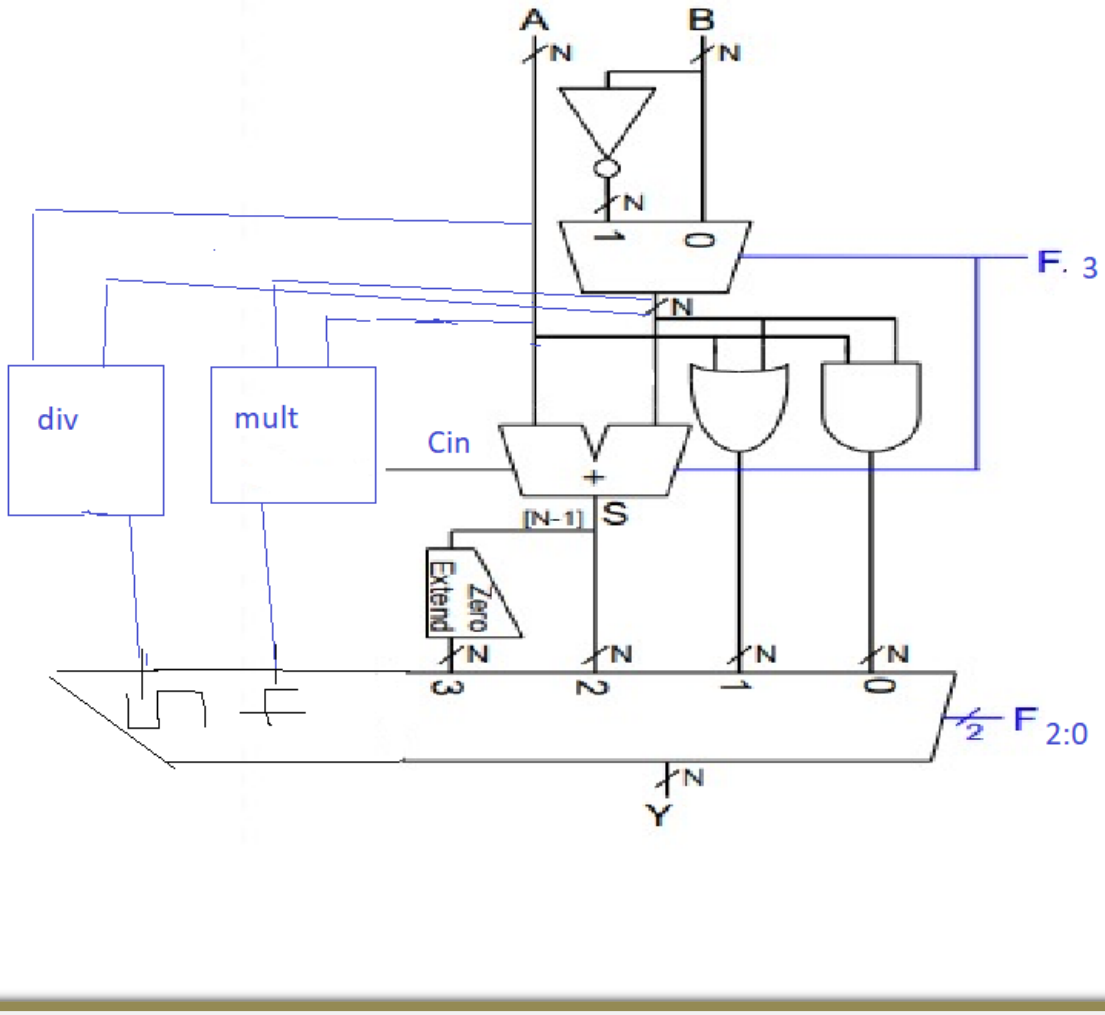
```
0x20100007
0x20110001
0x20120000
0x20130003
0x02119020
0x2252FFFF
0x02539018
0x0251901A
```



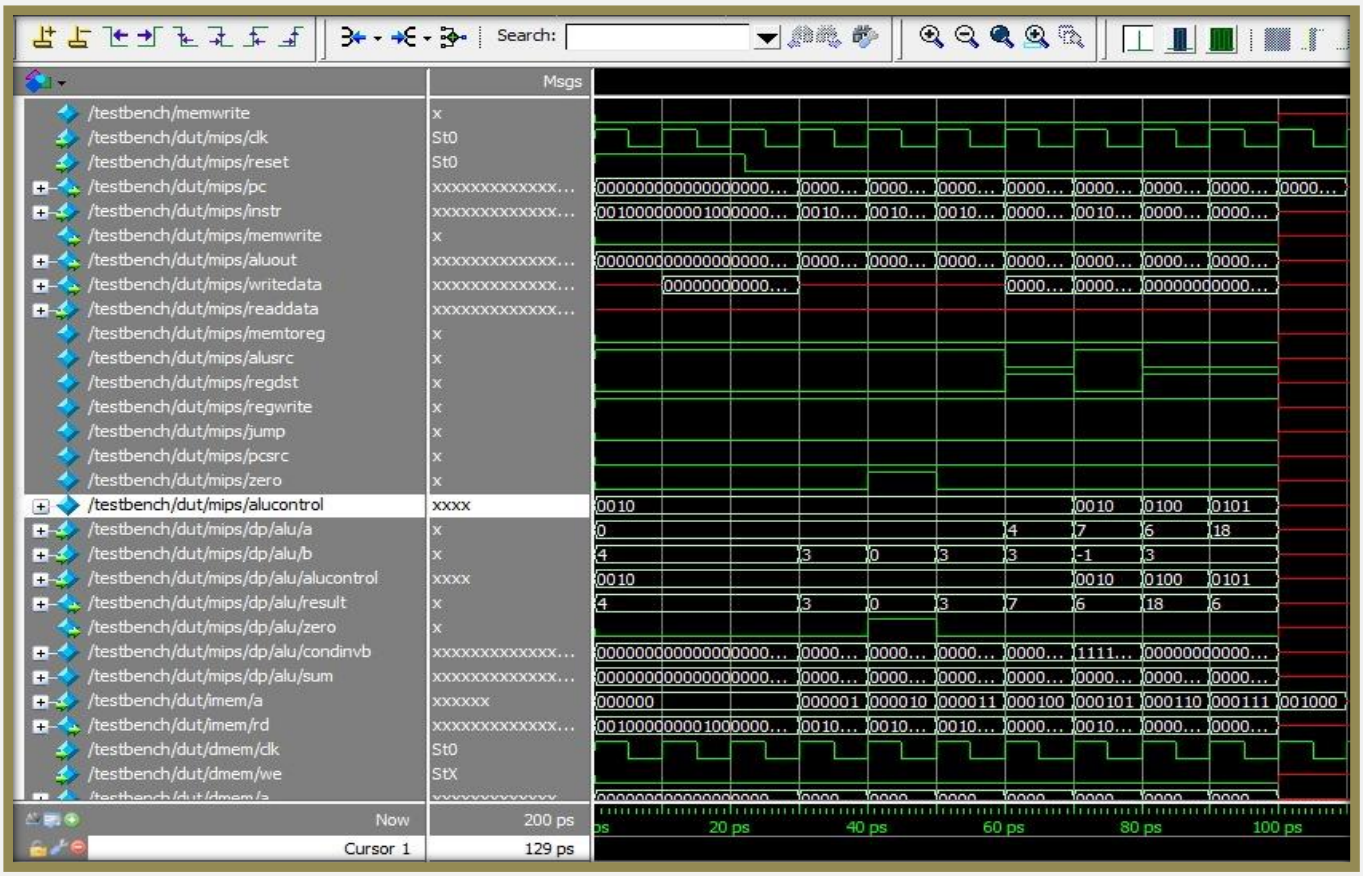
Text Segment

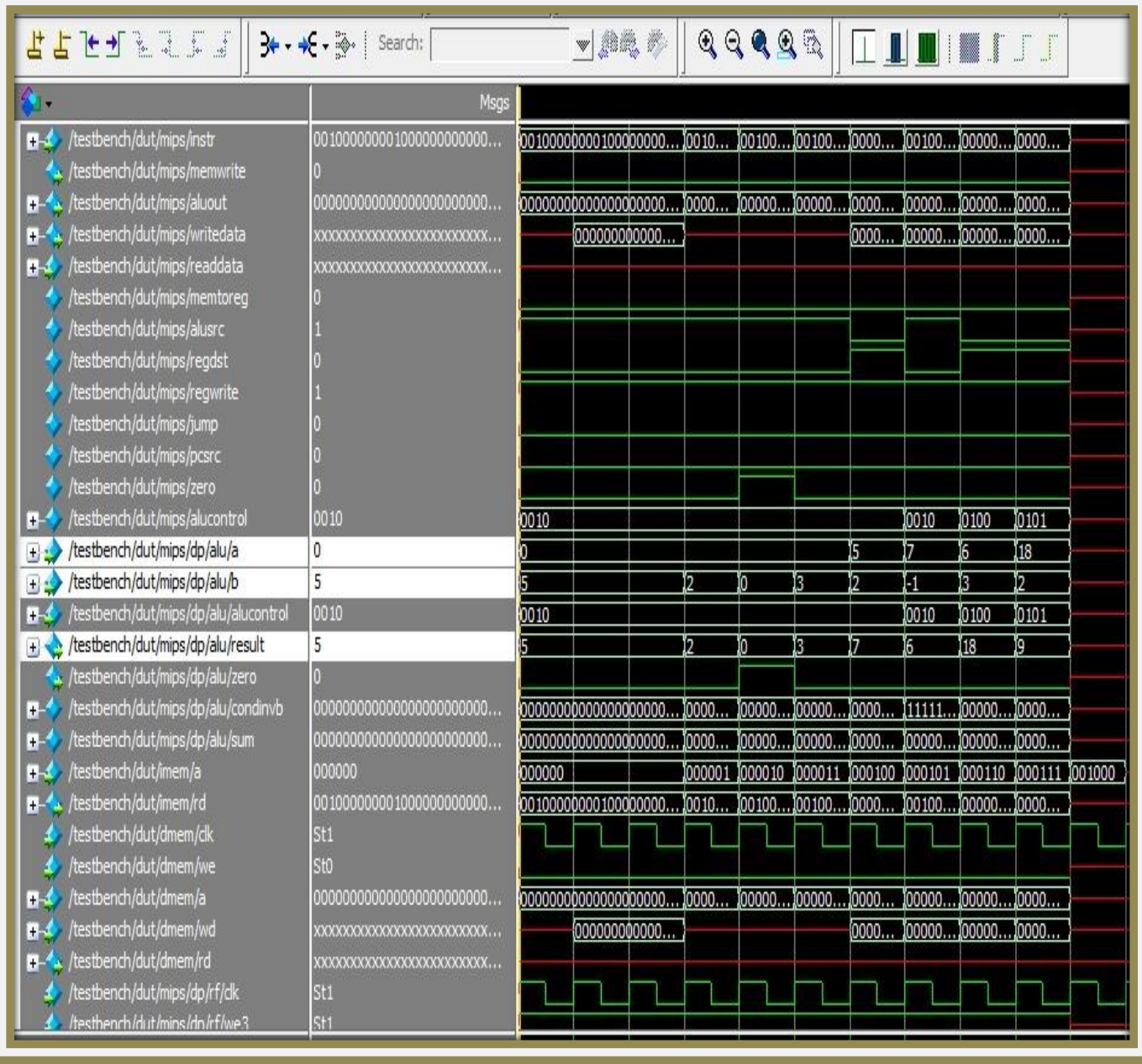
Bkpt	Address	Code	Basic	
	0x00400000	0x20100007	addi $16,$0,0x00000007	3: addi $s0,$zero,7
	0x00400004	0x20110001	addi $17,$0,0x00000001	4: addi $s1,$zero,1
	0x00400008	0x20120000	addi $18,$0,0x00000000	5: addi $s2,$zero,0
	0x0040000c	0x20130003	addi $19,$0,0x00000003	6: addi $s3,$zero,3
	0x00400010	0x02119020	add $18,$16,$17	7: add $s2,$s0,$s1
	0x00400014	0x2252ffff	addi $18,$18,0xffff...	8: addi $s2,$s2,-1
	0x00400018	0x72539002	mul $18,$18,$19	9: mul $s2,$s2,$s3
	0x0040001c	0x16200001	bne $17,$0,0x00000001	10: div $s2,$s2,$s1
	0x00400020	0x0000000d	break	
	0x00400024	0x0251001a	div $18,$17	
	0x00400028	0x00009012	mflo $18	

➢ Our handle for ALU



➢ Test bensch for two cases




Search:
Msgs
/testbench/dut/mips/instr
/testbench/dut/mips/memwrite
/testbench/dut/mips/aluout
/testbench/dut/mips/writedata
/testbench/dut/mips/readdata
/testbench/dut/mips/memtoreg
/testbench/dut/mips/alusrc
/testbench/dut/mips/regdst
/testbench/dut/mips/regwrite
/testbench/dut/mips/jump
/testbench/dut/mips/pcsrc
/testbench/dut/mips/zero
/testbench/dut/mips/alucontrol
/testbench/dut/mips/dp/alu/a
/testbench/dut/mips/dp/alu/b
/testbench/dut/mips/dp/alu/alucontrol
/testbench/dut/mips/dp/alu/result
/testbench/dut/mips/dp/alu/zero
/testbench/dut/mips/dp/alu/condinvb
/testbench/dut/mips/dp/alu/sum
/testbench/dut/imem/a
/testbench/dut/imem/rd
/testbench/dut/dmem/clk
/testbench/dut/dmem/we
/testbench/dut/dmem/a
/testbench/dut/dmem/wd
/testbench/dut/dmem/rd
/testbench/dut/mips/dp/rf/clk
/testbench/dut/mips/dp/rf/we3