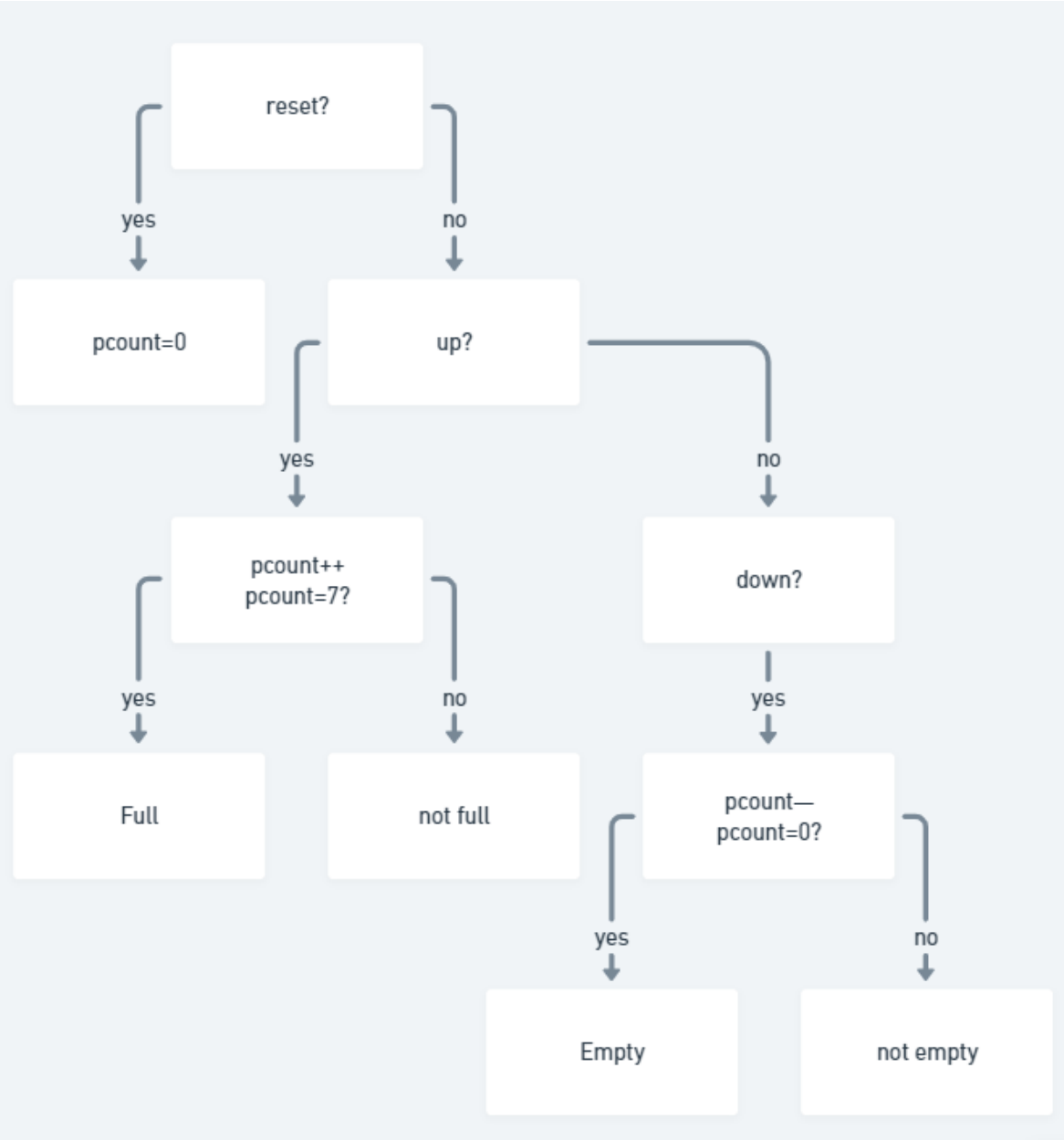
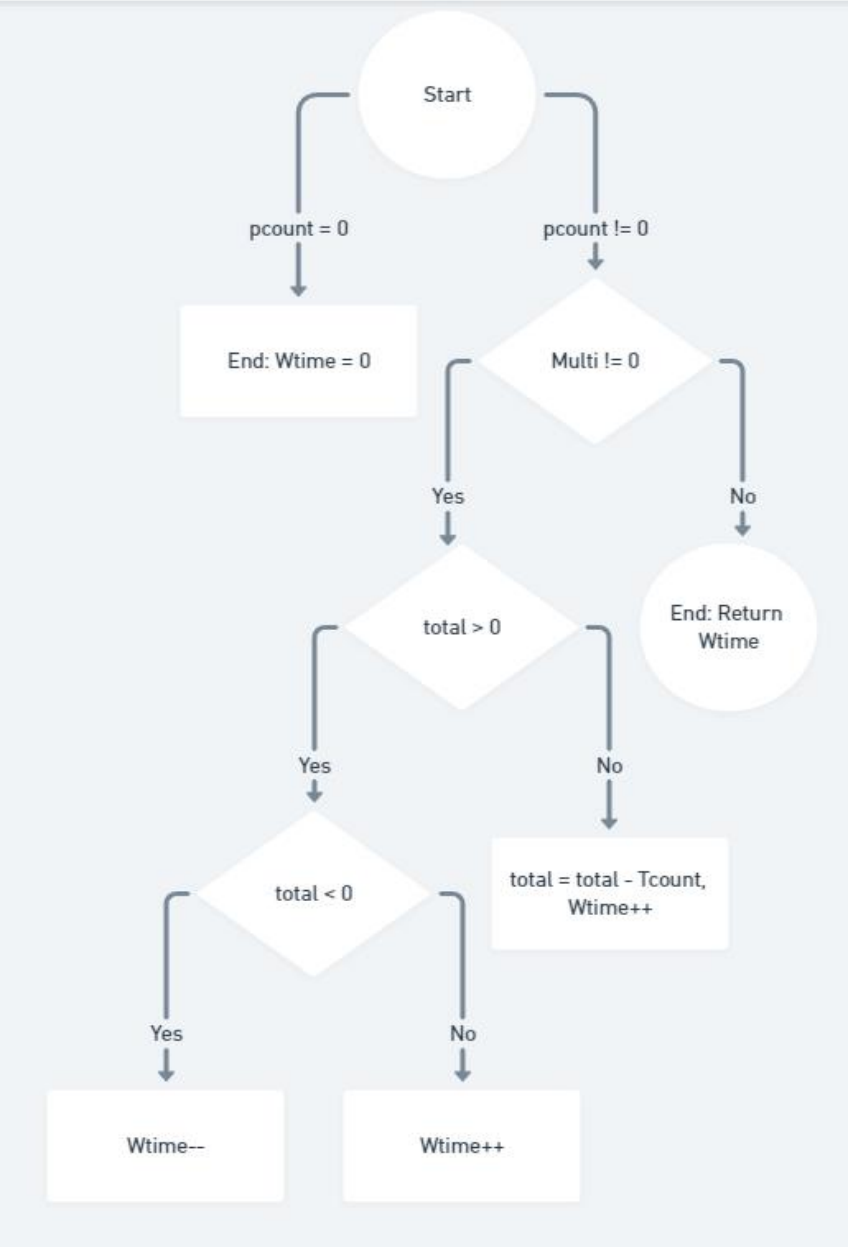
Final-Project

The manager of the ALameinBank bank branch, located in a Alamein is proposing to install an embedded system to monitor the client queue in front of the tellers. The proposed system, called ABQM™, is to display various information about the status of the queue.

Input/Output	Description	Possible Values
up	Signal indicating if the back-end queue is clear	Logic '0' (interrupted) or '1' (clear)
down	Signal indicating if the front-end queue is clear	Logic '0' (interrupted) or '1' (clear)
Pcount	Number of people standing in the queue	Integer, incremented or decremented by 1
Tcount	Number of tellers currently in service	Integer, 1, 2, or 3
Reset	Signal to reset the system	Logic '1' (reset) or '0' (normal)
Empty flag	Flag indicating if the queue is empty	Logic '1' (empty) or '0' (not empty)
Full flag	Flag indicating if the queue is full	Logic '1' (full) or '0' (not full)
Wtime	Expected waiting time in the queue before being served	Integer, calculated based on Pcount and Tcount


reset?
yes
no
pcount=0
up?
yes
no
pcount++
pcount=7?
down?
yes
no
yes
Full
not full
pcount—
pcount=0?
yes
no
Empty
not empty


Start
pcount = 0
pcount != 0
End: Wtime = 0
Multi != 0
Yes
No
End: Return Wtime
total > 0
Yes
No
total = total - Tcount, Wtime++
total < 0
Yes
No
Wtime--
Wtime++

- Model the up-down counter as a separate function using C language.

```c
#include <stdio.h>
#include <stdbool.h>

int up_downCounter(bool reset, bool up, bool down, int pcount) {
    if(reset) {
        pcount = 0;
    } else if(up && pcount != 7) {
        pcount++;
    } else if(down && pcount != 0) {
        pcount--;
    }
    return pcount;
}

```

- Model the required functions in C language.

```c
        pcount++;
    } else if(down && pcount != 0) {
        pcount--;
    }
    return pcount;
}

void flag(int pcount, bool *empty, bool *full) {
    if(pcount == 0) {
        *empty = true;
        *full = false;
    } else if(pcount == 7) {
        *empty = false;
        *full = true;
    } else {
        *empty = false;
        *full = false;
    }
}

int Wtime_Calculations(int pcount, int Tcount) {
    int Wtime = 0;
    int total = 0;
    int Multi = 3;
    if (pcount == 0) {
        Wtime = 0;
    } else {
        while (Multi != 0) {
```

```c
}

int Wtime_Calculations(int pcount, int Tcount) {
    int Wtime = 0;
    int total = 0;
    int Multi = 3;
    if (pcount == 0) {
        Wtime = 0;
    } else {
        while (Multi != 0) {
            total = total + (pcount + Tcount - 1);
            Multi--;
        }
        while (total > 0) {
            total = total - Tcount;
            Wtime++;
        }
        if (total < 0) {
            Wtime--;
        }
    }
    return Wtime;
}

int main() {
    int pcount = 0;
    bool empty = false, full = false;
```

```
        }
        if (total < 0) {
            Wtime--;
        }
    }
    return Wtime;
}

int main() {
    int pcount = 0;
    bool empty = false, full = false;
    int Tcount = 5; // Example value for Tcount

    for(int i = 0; i <= 7; i++) {
        printf("Count: %d\n", pcount);
        flag(pcount, &empty, &full);
        printf("Full: %d, Empty: %d\n", full, empty);
        pcount = up_downCounter(false, true, false, pcount);


    }

    // Example usage of Wtime_Calculations
    int Wtime = Wtime_Calculations(pcount, Tcount);
    printf("Wtime: %d\n", Wtime);

    return 0;
}
```

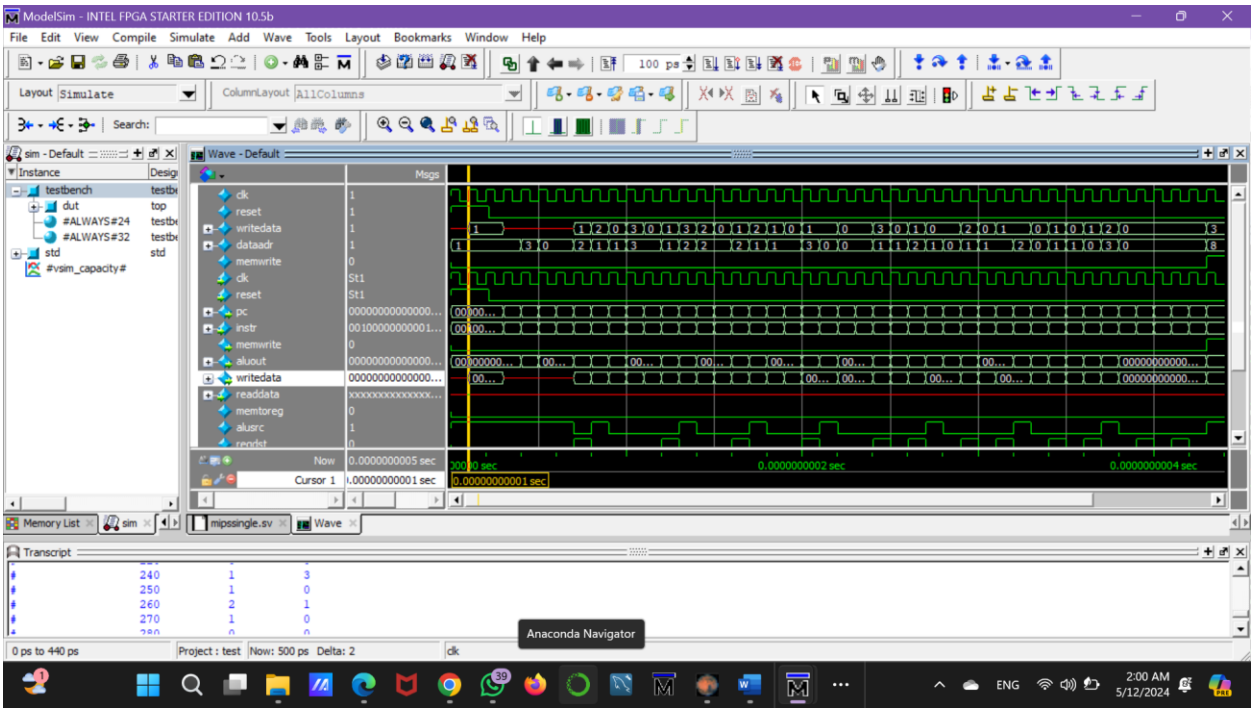
- Translate the C function Wtime into MIPS assembly language.

```
1  #pcount=$a0   Tcount=$a1  multi=$s0  total=$s1
2  # wtime=$t0  pcount+Tcount-1=$s2
3  addi $a0,$0,3  #pcount=3
4  addi $a1,$0,2  #Tcount=2
5  addi $s0 ,$0,3 #multi=3
6  addi $s1,$0,0   #total=0
7  addi $t0,$0,0   #wtime=0
8  add $s2,$a0,$a1  #pcount+Tcount
9  addi $s2,$s2,-1  # pcount+Tcount-1
10 beq $a0,$0,done  #if (pcount=0)
11
12
13 j multi
14
15 multi:
16 beq $s0,$0,divide #while (Multi != 0)
17 add $s1,$s1,$s2 #total=total+pcount+Tcount-1
18 addi $s0,$s0,-1 # Multi=Multi-1
19 j multi
20
21
22 divide:
23 slt $t1,$0,$s1 #while (total > 0)
24 beq $t1,$0,divide2
25 sub $s1,$s1,$a1  #total=total-Tcount
```

```
13  j multi
14
15  multi:
16  beq $s0,$0,divide #while (Multi != 0)
17  add $s1,$s1,$s2 #total=total+pcount+Tcount-1
18  addi $s0,$s0,-1 # Multi=Multi-1
19  j multi
20
21
22  divide:
23  slt $t1,$0,$s1 #while (total > 0)
24  beq $t1,$0,divide2
25  sub $s1,$s1,$a1  #total=total-Tcount
26  addi $t0,$t0,1   #wtime=wtime+1
27  j divide
28
29
30  divide2:
31  slt $t2,$s1,$0  #while(total < 0)
32  beq $t2,$0,done
33  addi $t0,$t0,-1 # wtime=wtime-1
34
35  done:
36  sw $t0,8($0)
```

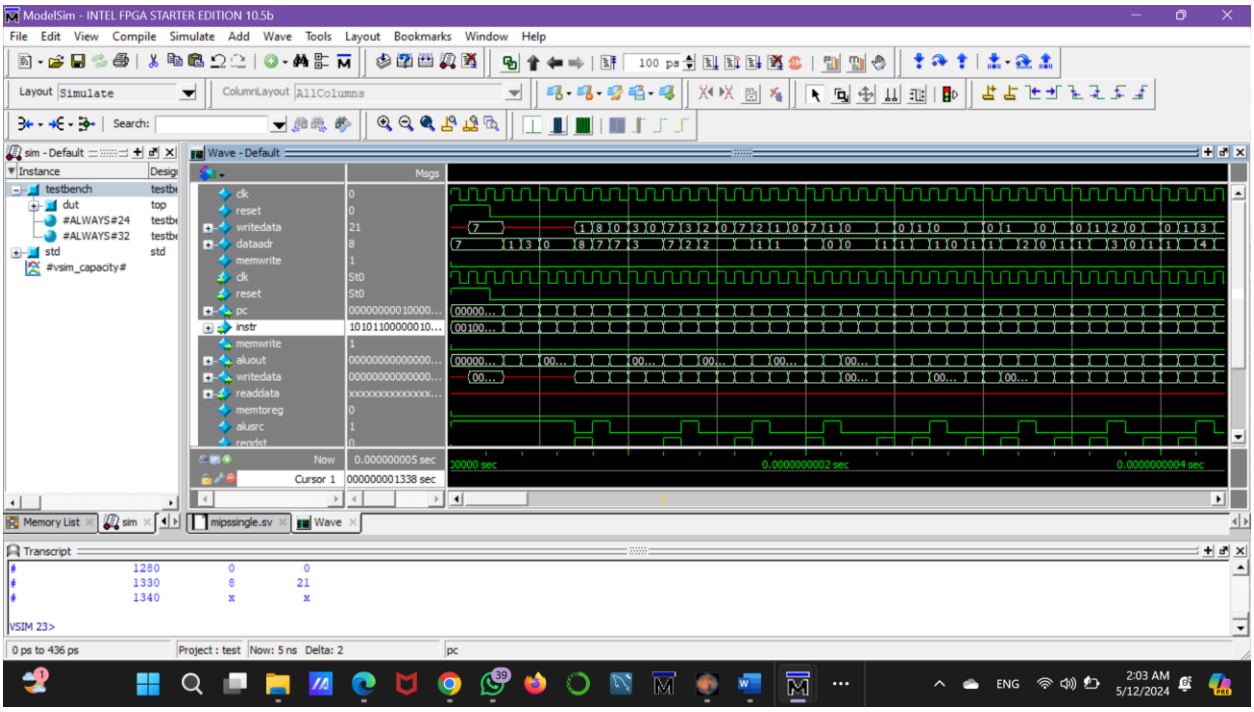
- Tcount=1 ,Pcount=1

Address	Code
0x00400000	0x20040001
0x00400004	0x20050001
0x00400008	0x20100003
0x0040000c	0x20110000
0x00400010	0x20080000
0x00400014	0x00859020
0x00400018	0x2252ffff
0x0040001c	0x1080000d
0x00400020	0x08100009
0x00400024	0x12000003
0x00400028	0x02328820
0x0040002c	0x2210ffff
0x00400030	0x08100009
0x00400034	0x0011482a
0x00400038	0x11200003
0x0040003c	0x02258822
0x00400040	0x21080001
0x00400044	0x0810000d
0x00400048	0x0220502a
0x0040004c	0x11400001
0x00400050	0x2108ffff
0x00400054	0xac080008



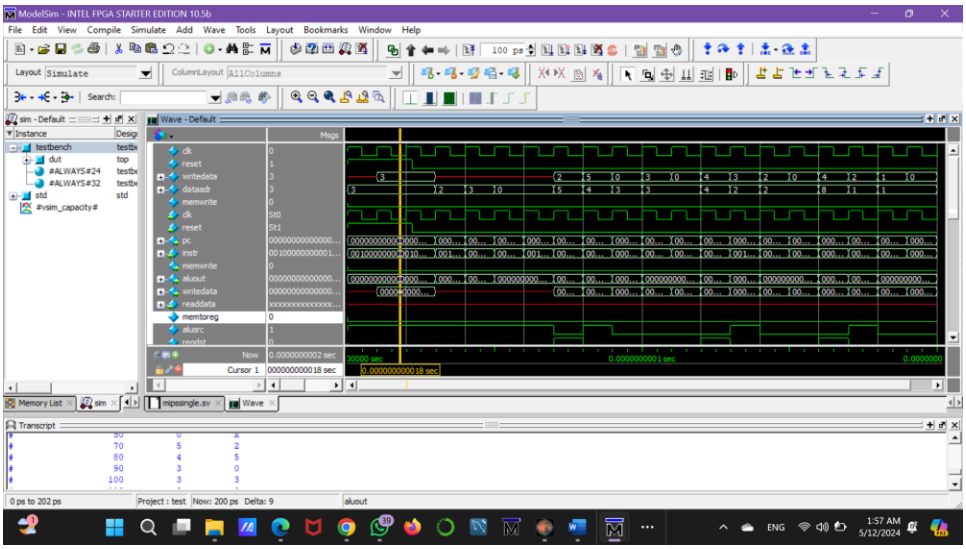
- Tcount=1 ,Pcount=7

Address	Code
0x00400000	0x20040007
0x00400004	0x20050001
0x00400008	0x20100003
0x0040000c	0x20110000
0x00400010	0x20080000
0x00400014	0x00859020
0x00400018	0x2252ffff
0x0040001c	0x1080000d
0x00400020	0x08100009
0x00400024	0x12000003
0x00400028	0x02328820
0x0040002c	0x2210ffff
0x00400030	0x08100009
0x00400034	0x0011482a
0x00400038	0x11200003
0x0040003c	0x02258822
0x00400040	0x21080001
0x00400044	0x0810000d
0x00400048	0x0220502a
0x0040004c	0x11400001
0x00400050	0x2108ffff
0x00400054	0xac080008



- Tcount=2 ,Pcount=3

Address	Code
0x00400000	0x20040003
0x00400004	0x20050002
0x00400008	0x20100003
0x0040000c	0x20110000
0x00400010	0x20080000
0x00400014	0x00859020
0x00400018	0x2252ffff
0x0040001c	0x1080000d
0x00400020	0x08100009
0x00400024	0x12000003
0x00400028	0x02328820
0x0040002c	0x2210ffff
0x00400030	0x08100009
0x00400034	0x0011482a
0x00400038	0x11200003
0x0040003c	0x02258822
0x00400040	0x21080001
0x00400044	0x0810000d
0x00400048	0x0220502a
0x0040004c	0x11400001
0x00400050	0x2108ffff
0x00400054	0xac080008



Made by : Malak yasser aly 22100400

Omar Ashraf 22102210

Fares Hassan Hamed 22101096