**Problem Definition:** Evolve a static formal methodology to estimate the worst case time taken by the given embedded C program to service the WatchDogTimer(WDT), taking into account the platform semantics.

**Application Program outline**

main()

{

WDT Initializations(); //Watch Dog Timer Initialisations

while(1)

{

WDT\_Servicing(); // triggering WDT

if(Data is received from Environment) // from a 155B bus connected to it

{

Switch(data)

{

Case a:

Statement 1; //these statements can involve access to hardware or

Statement 2; some logical computations

….

….

break;

Case b:

Statement 1;

Statement 2;

….

….

break;

…..

…..

Default:

break;

} //end of switch statement

} //end of if

} //end of while(1)

}//end of program

**Approach:**

**Hardware**

Interface Layer

Application Program

Interface Layer--

Hardware model of a WatchDog Timer: It is nothing but a down counter. Countdown timer decreases its contents by 1 each clock. When 0 is reached, interrupt is asserted.

It has following register set

* A register to enable the WDT
* Register which configures the clock input to the counter
* Count register to load the count value to the counter.

counter value

Counter