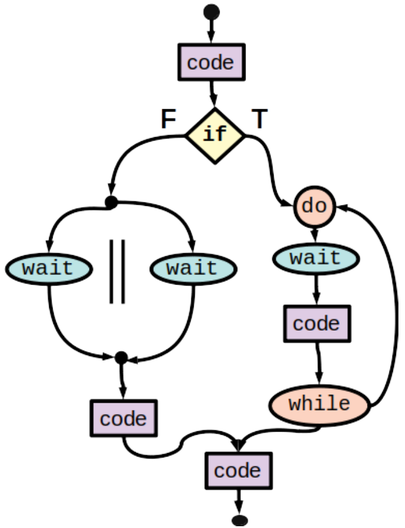
```
module Top
  behavior
    decl $int x;$;
    //Input x from user
    $ std::cout<<"\nEnter a number:";std::cin>>x;$;
    //Branch
    if($x%2==0$) then
        do //Loop
            wait(1,0); //wait for one cycle
            $log<<endl<<"In the even branch";$;
        while($current_time.cycle()<4$) end do;
        stop simulation;
    else
        [  //Parallel block
            wait (2,0);
        ||
            wait until ($current_time>=time(x,0)$);
        ];
        $log<<endl<<"In the odd branch";$;
    end if;
    $log<<endl<<"Done";$;
    stop simulation;
  end behavior
end module
```

(A) Example illustrating a module's behavioral description



Output (with an even number as input)

```
Enter a number:2

(1,0)TOP        :IN the even branch
(2,0)TOP        :IN the even branch
(3,0)TOP        :IN the even branch
(4,0)TOP        :IN the even branch
(4,0)TOP        :Done
Simulation stopped at time (4,0)
```

Output (with an odd number as input)

```
Enter a number:5

(5,0)TOP        :In the odd branch
(5,0)TOP        :Done
Simulation stopped at time (5,0)
```

(A) Control flow and generated outputs

```
//A Shift register model
module Top
    submodule S : ShiftRegister<3,2>
    //Instantiate a shift register with <num_stages=3,delay=2>
end module

module ShiftRegister
    parameter int N = 1     //number of stages
    parameter int DELAY = 1 //delay of each stage

    submodule p : Producer
    submodule c : Consumer
    submodule_array stage[N]  : Stage<DELAY>
    net_array n[N+1] : capacity 1

    //make connections
    for i in 0 to (N - 1)
        stage[i].ip <= n[i]
        stage[i].op => n[i+1]
    end for
    p.op => n[0]
    c.ip <= n[N]
end module
```

(B) Example (partial) illustrating structural descriptions in Sitar