Exam #2: Train Crossing Controller

- Quiz #2, revisited, with fewer simplifying assumptions and concurrent SynchSMs

I/O Description

<u>Inputs</u>:

A0	Detect that a train has entered the crossing zone
A1	Detect that a train has exited the crossing zone
A2	Detect that the crossbar is fully lowered
A3	Detect that the crossbar is fully raised
A5-A4	Speed of the train
	00: Slow
	01: Medium
	10: Fast
	11: Shinkansen (Japanese Bullet Train)

<u>Outputs</u>:

B0	PWM signal to raise/lower the crossbar
B1	Crossbar Direction
	0: Lower the crossbar
	1: Raise the crossbar
B2	Light Control

System Functionality (1/5)

- When a train enters the crossing zone (A0 = 1 for 100 ms) a sensor detects the velocity of the train (A5-A4) and lowers the crossbar (B1 = 0).
- A PWM signal on B0 controls the rotational velocity of the servo that lowers the crossbar:
  - Velocity = Low: 20% duty cycle
  - Velocity = Medium: 40% duty cycle
  - Velocity = High: 60% duty cycle
  - Velocity = Shinkansen: 80% duty cycle

System Functionality (2/5)

- The PWM signal to lower the crossbar stops (B0 = 0) when sensor A2 detects that the crossbar is fully lowered (A2 = 1).
- When the train exits the crossing zone (A1 = 1 for 100 ms) the system raises the crossbar (B1 = 1) using a PWM signal on B0 with a 50% duty cycle.
- The PWM signal stops (B0 = 0) when sensor A3 detects that the crossbar is fully raised (A3 = 1).

System Functionality (3/5)

- The velocity of a train traveling through the crossing zone may change at any time. At any given time, the PWM duty cycle should reflect the current velocity.
-
- Multiple trains may travel through the crossing zone at a given time. When multiple trains are present, A5-A4 will report the velocity of the fastest traveling train in the zone. The velocity of the fastest train in the crossing zone determines the PWM duty cycle, as stated above.

System Functionality (4/5)

- The PWM signal that controls the servo motors must be shot off within one tick of detecting that the crossbar is fully lowered (A2 = 1) or fully raised (A3 = 1).
  - It is possible that the last train leaves the crossing zone before the crossbar is fully lowered
  - It is also possible that a new train enters an otherwise empty crossing zone before the crossbar is fully raised.

System Functionality (5/5)

- When the system is first turned on, it is guaranteed that no train will be in the crossing zone. The system includes sensors to detect trains entering/leaving the crossing zone; the system cannot directly sense the presence (or count the number) of trains in the crossing zone.

- The light is off (B2 = 0) when the crossbar is fully up. The light is on (B2 = 1) when the crossbar is fully down. The light blinks (500 ms on, 500 ms off) when the crossbar is being raised or lowered.

Directions and System Parameters

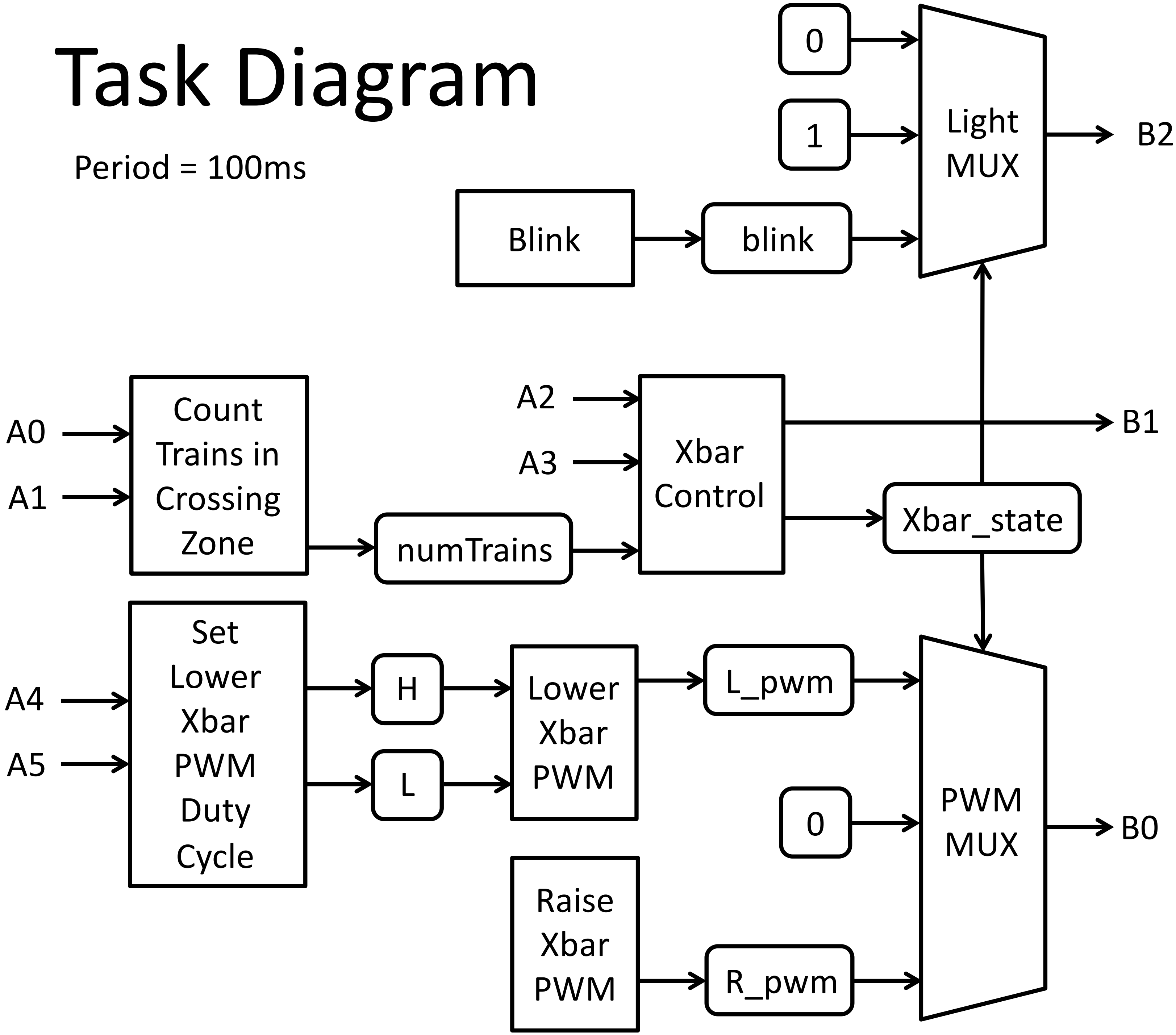
- Draw a task diagram with tasks, inputs, outputs, and shared variables clearly labeled. Remember to declare the type of each shared variable.

- Draw a SynchSM for each task in the system.

- PWM period: 1000 ms
- System period: 100 ms

Task Diagram

Period = 100ms

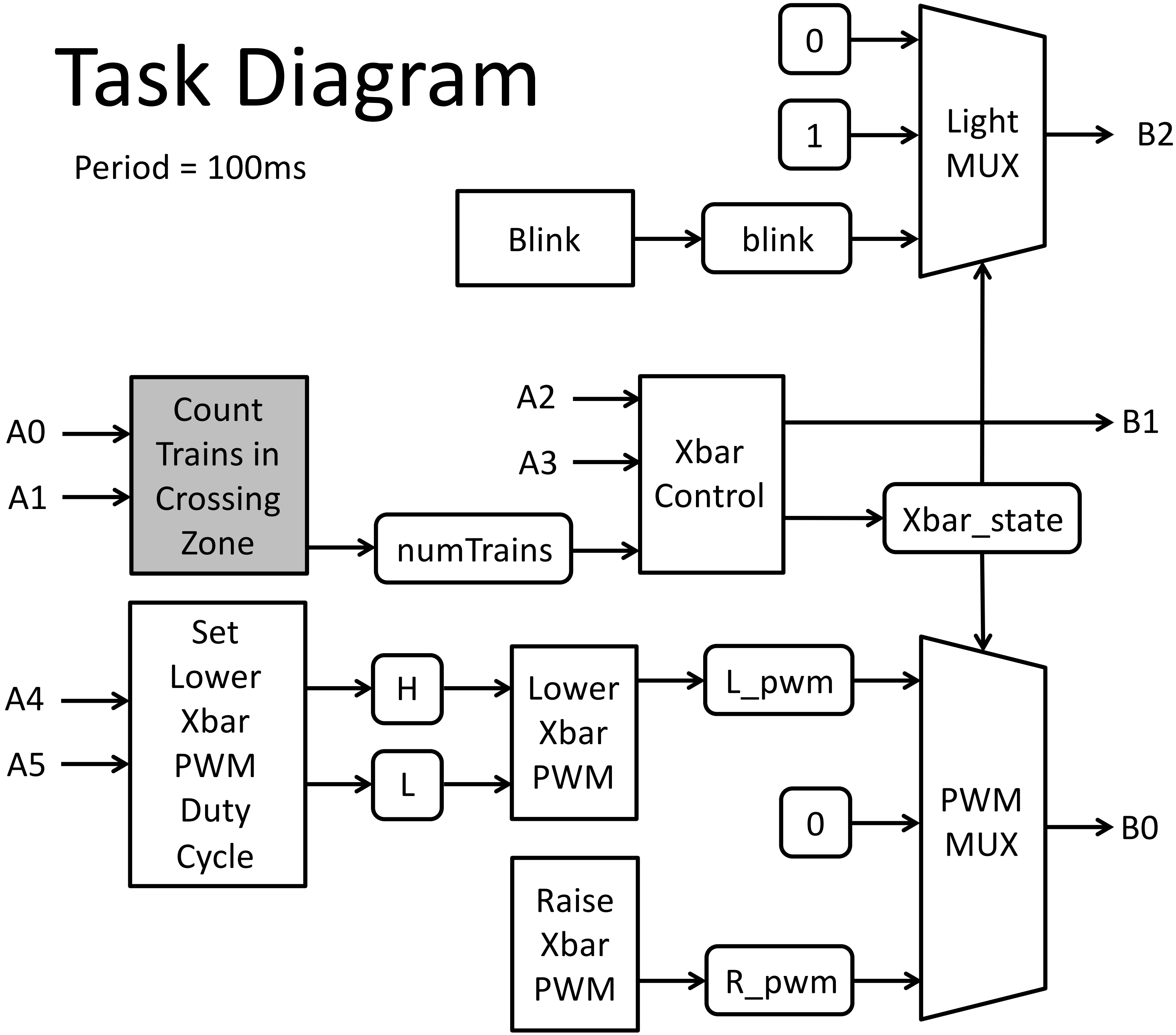


Shared Variable Declarations

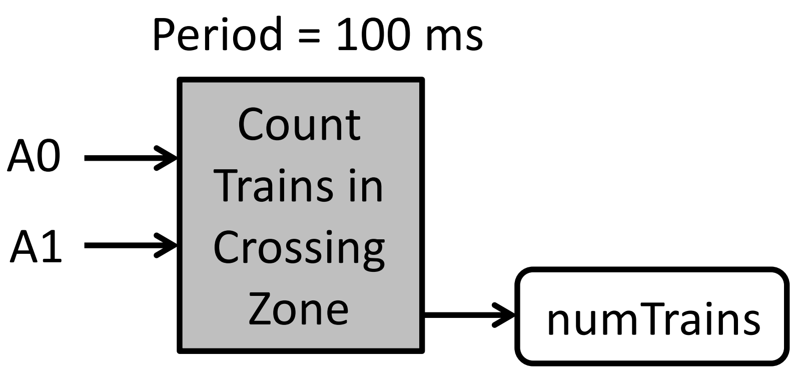
```
unsigned char numTrains;
unsigned char H, L;
unsigned char R_pwm, L_pwm;
unsigned char blink;
enum Xbar_States
    {Up, Down, GoingUp, GoingDown} xbar_state;
```

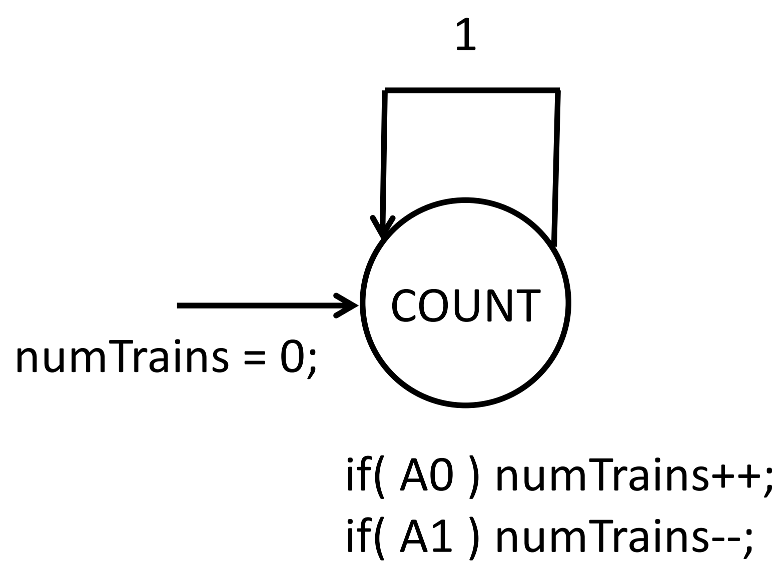
Task Diagram

Period = 100ms



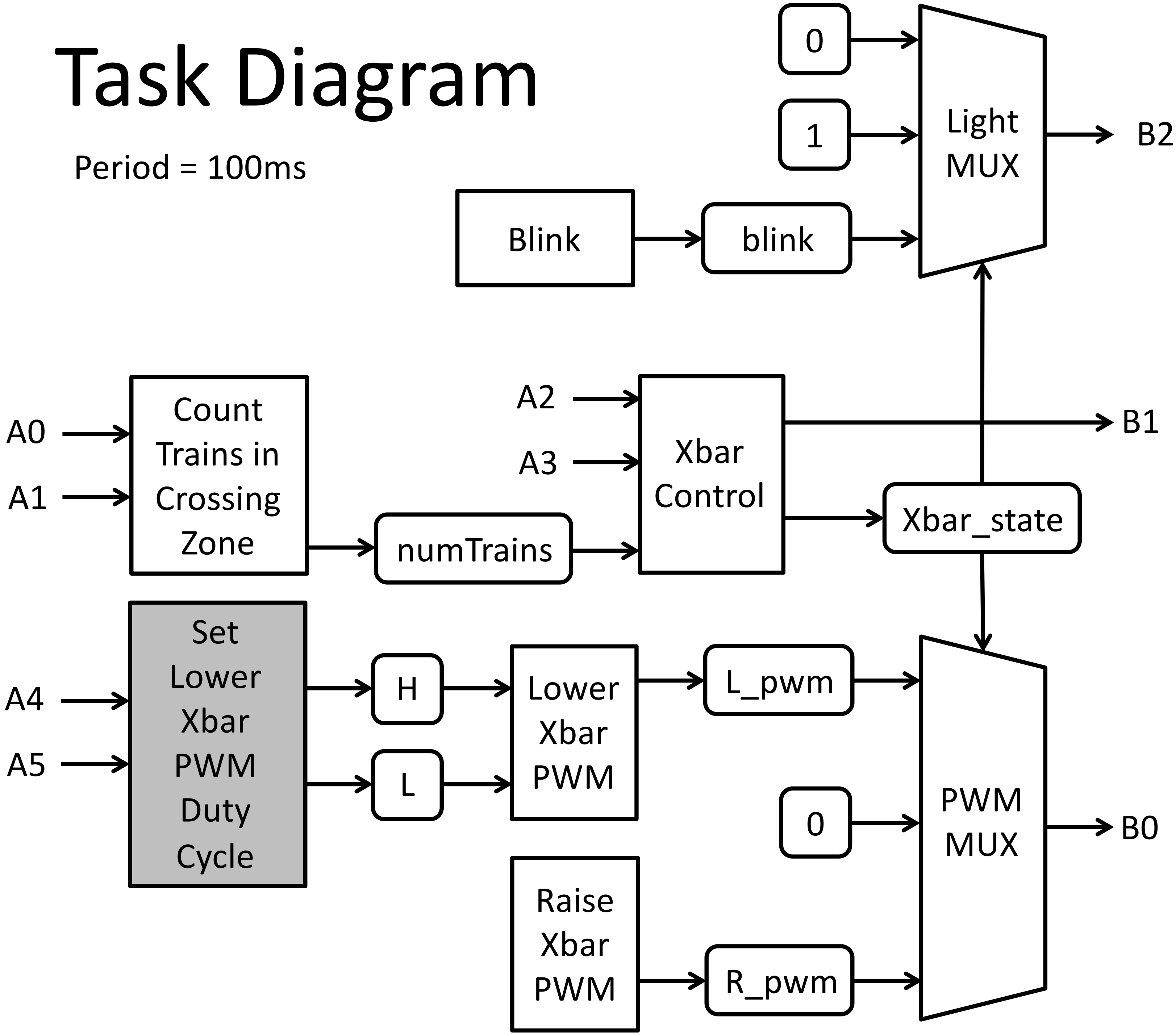
Count Trains in Crossing Zone



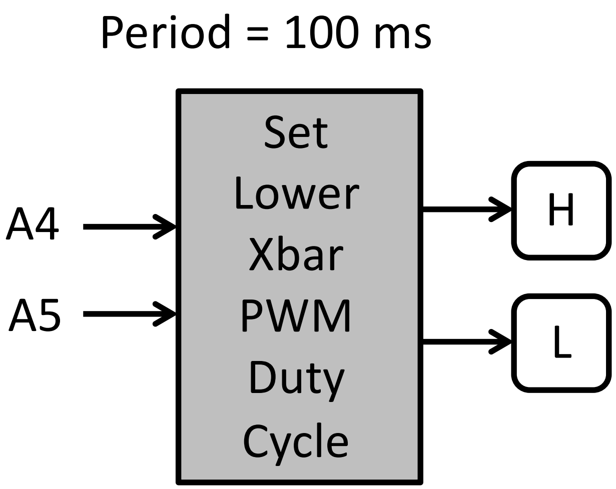


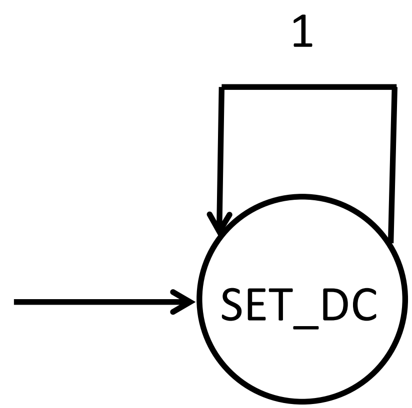
Task Diagram

Period = 100ms



Set Lower Xbar PWM Duty Cycle

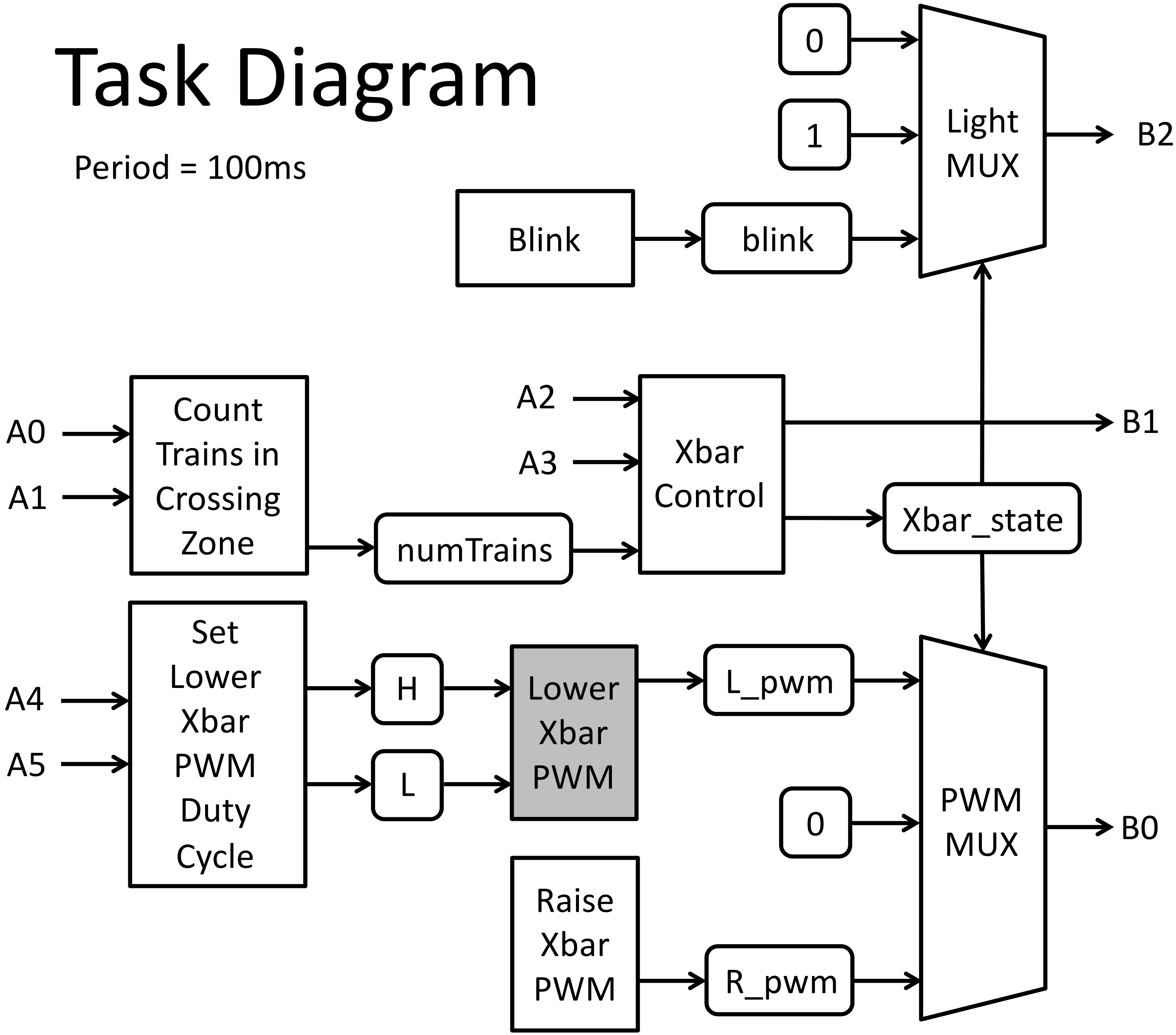




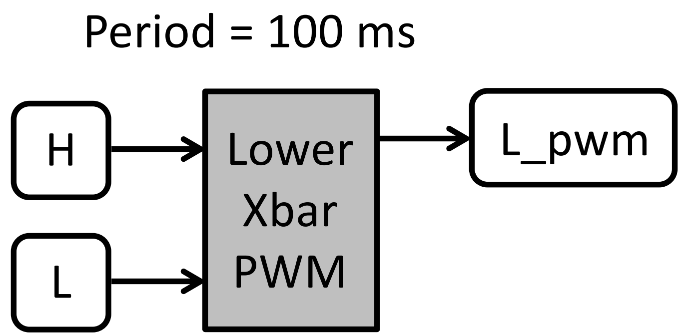
if($\overline{A5}$ && $\overline{A4}$) { H = 20; L = 80; }
else if($\overline{A5}$ && A4) { H = 40; L = 60; }
else if(A5 && $\overline{A4}$) { H = 60; L = 40; }
else { H = 80; L = 20; }

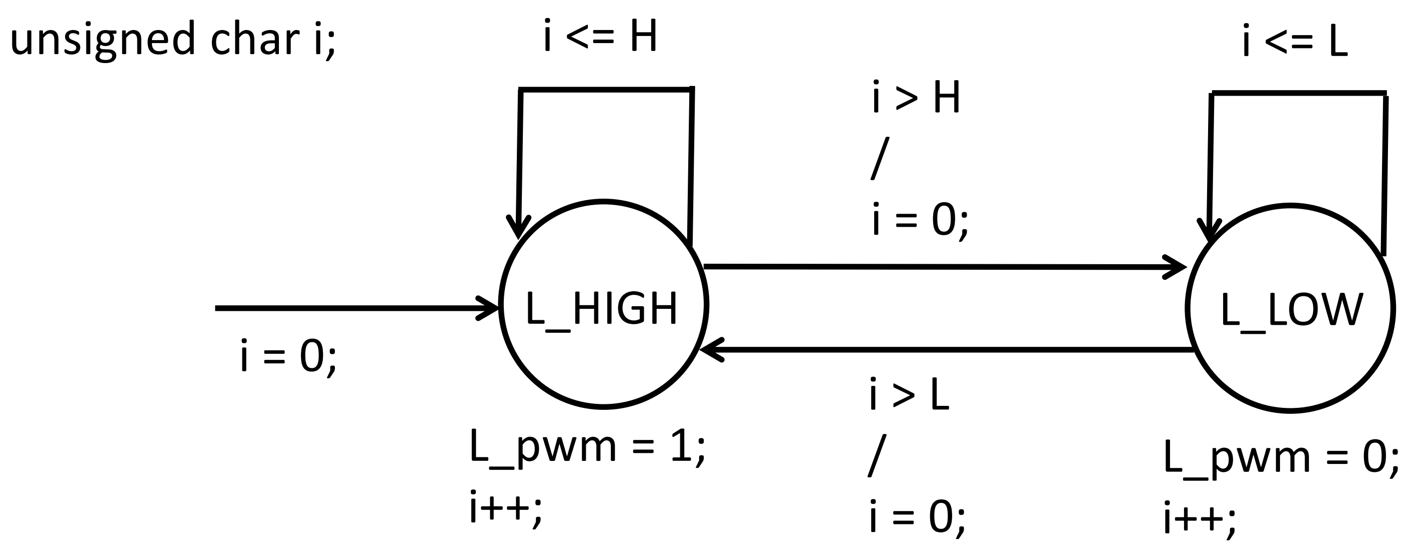
Task Diagram

Period = 100ms



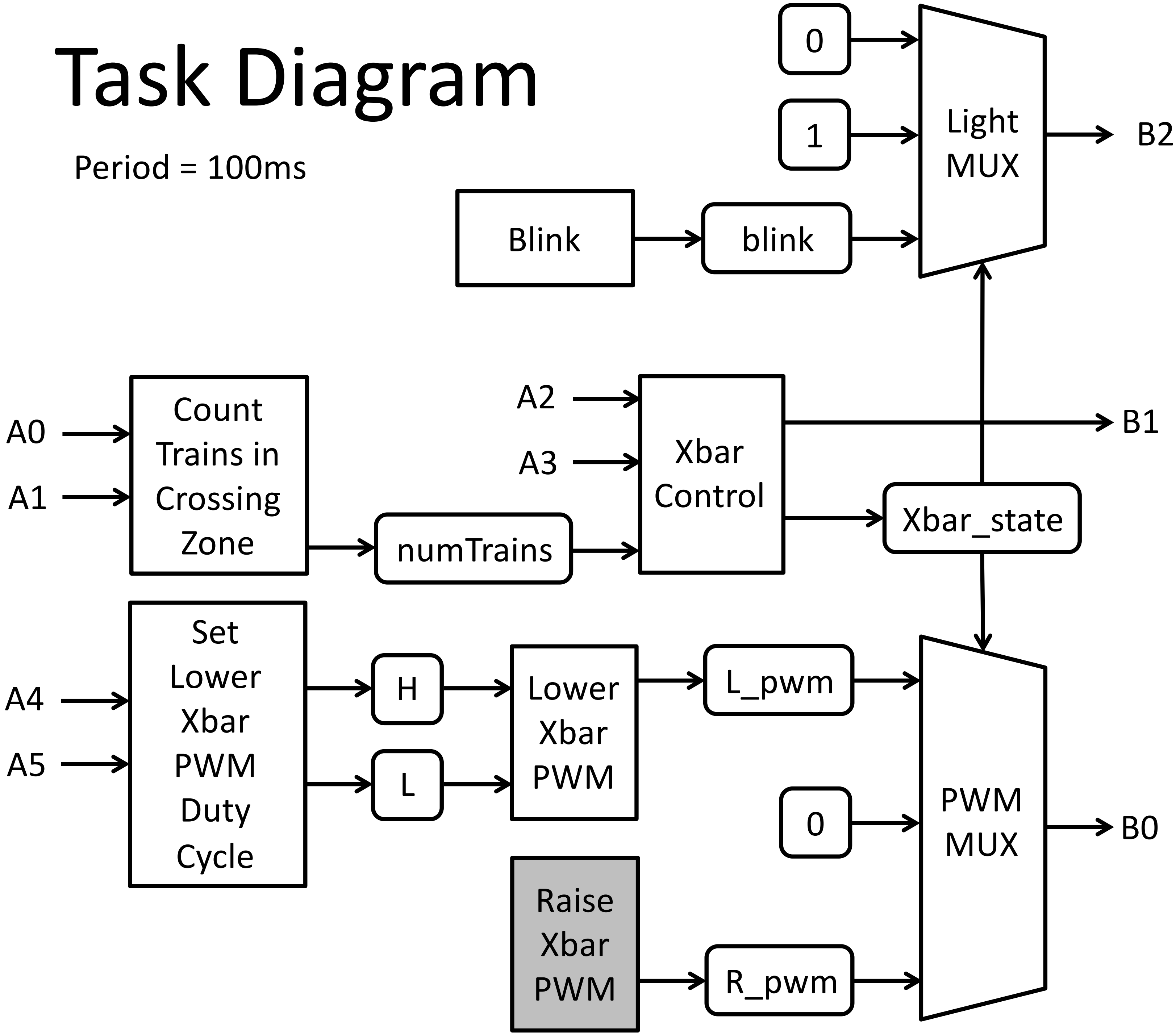
Lower Xbar PWM



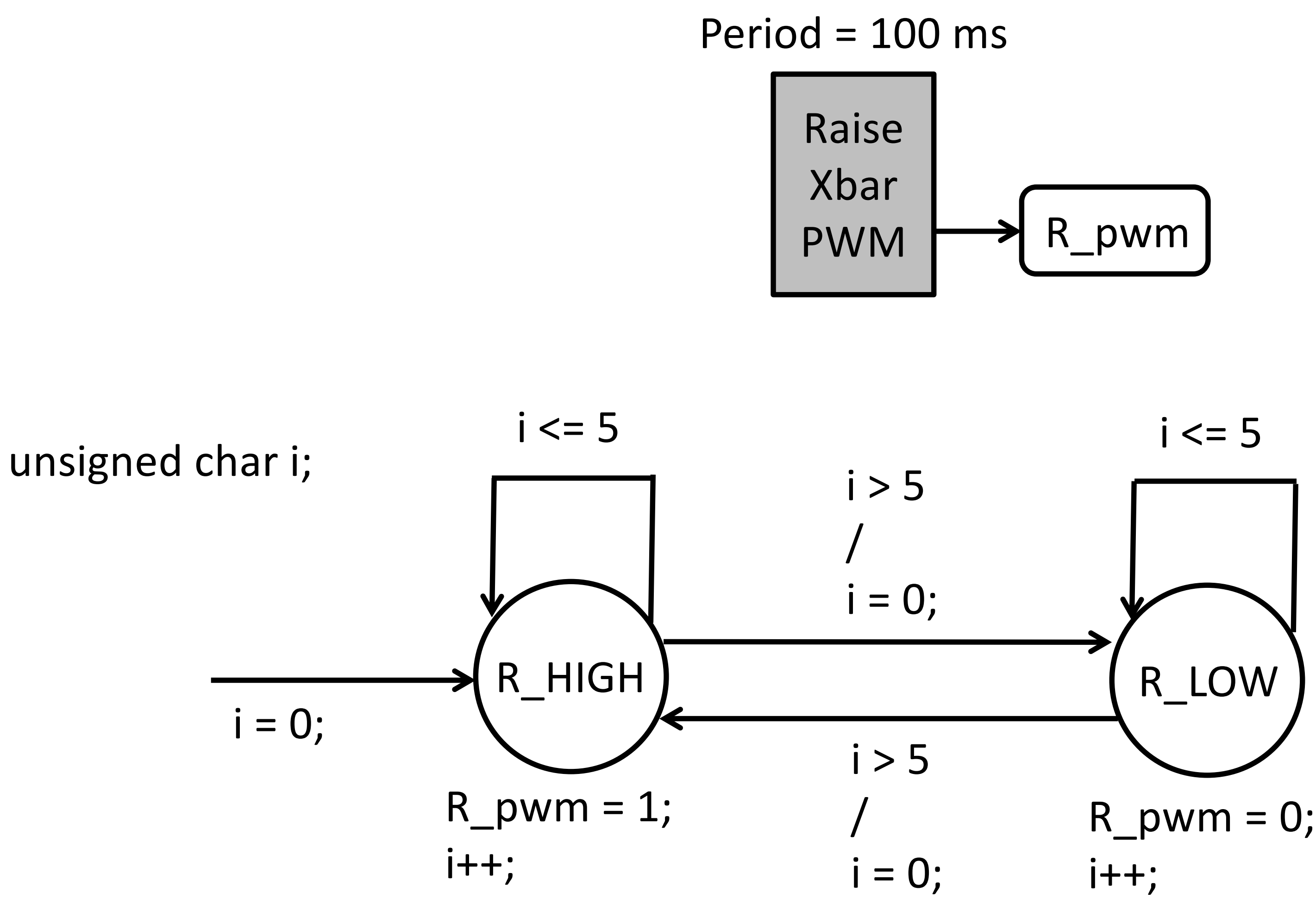


Task Diagram

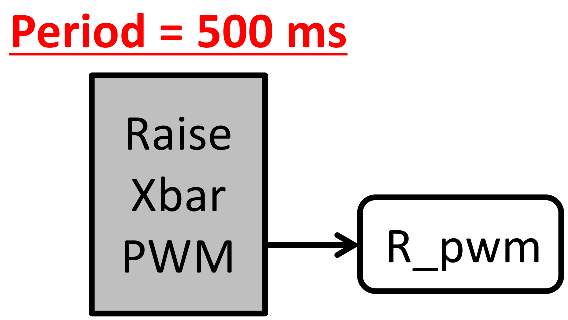
Period = 100ms

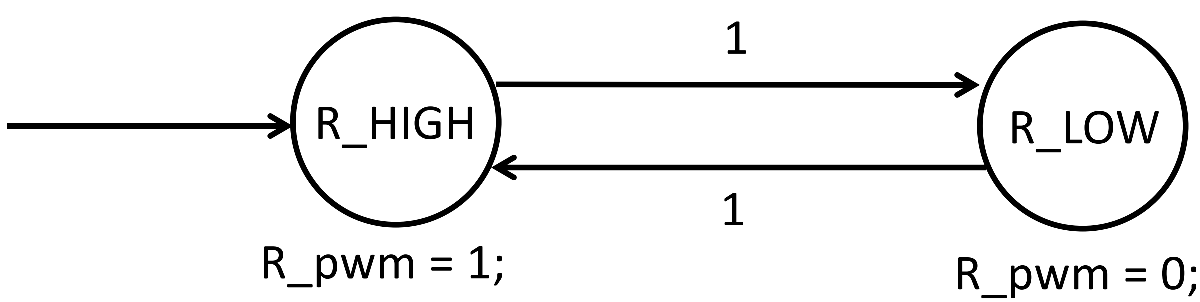


Raise Xbar PWM



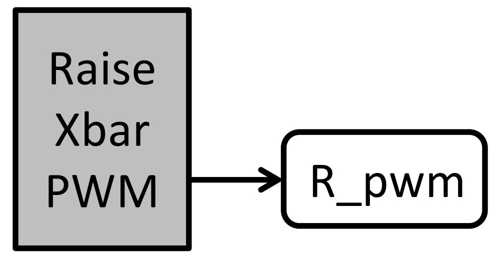
Raise Xbar PWM (Simpler Solution)



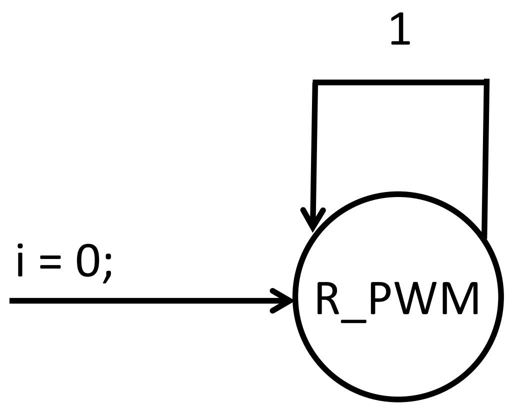


Raise Xbar PWM (Simpler Solution)

Period = 500 ms



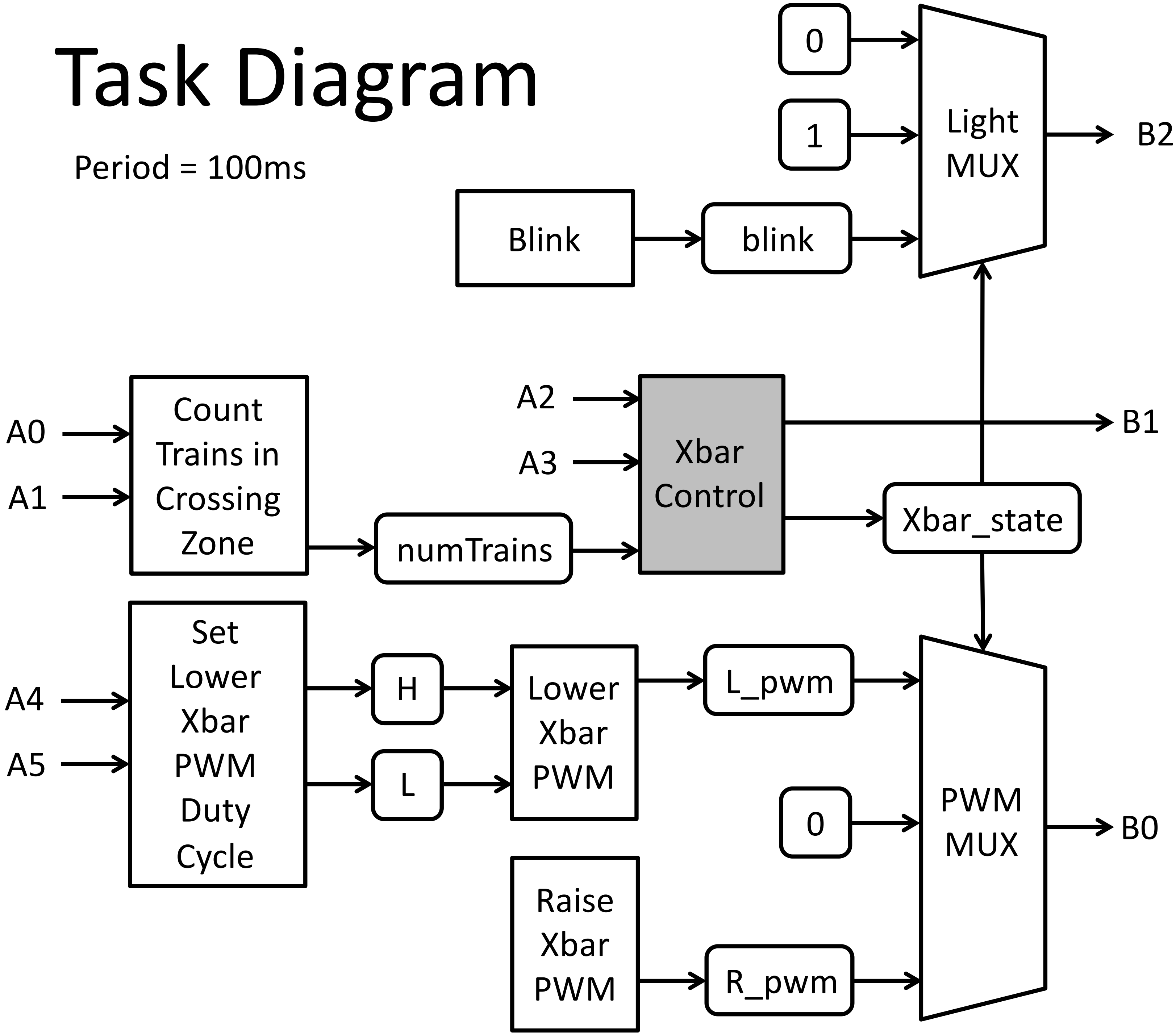
unsigned char i;




```
i = (i+1)%2;
R_pwm = (i == 0) ? 0 : 1;
```

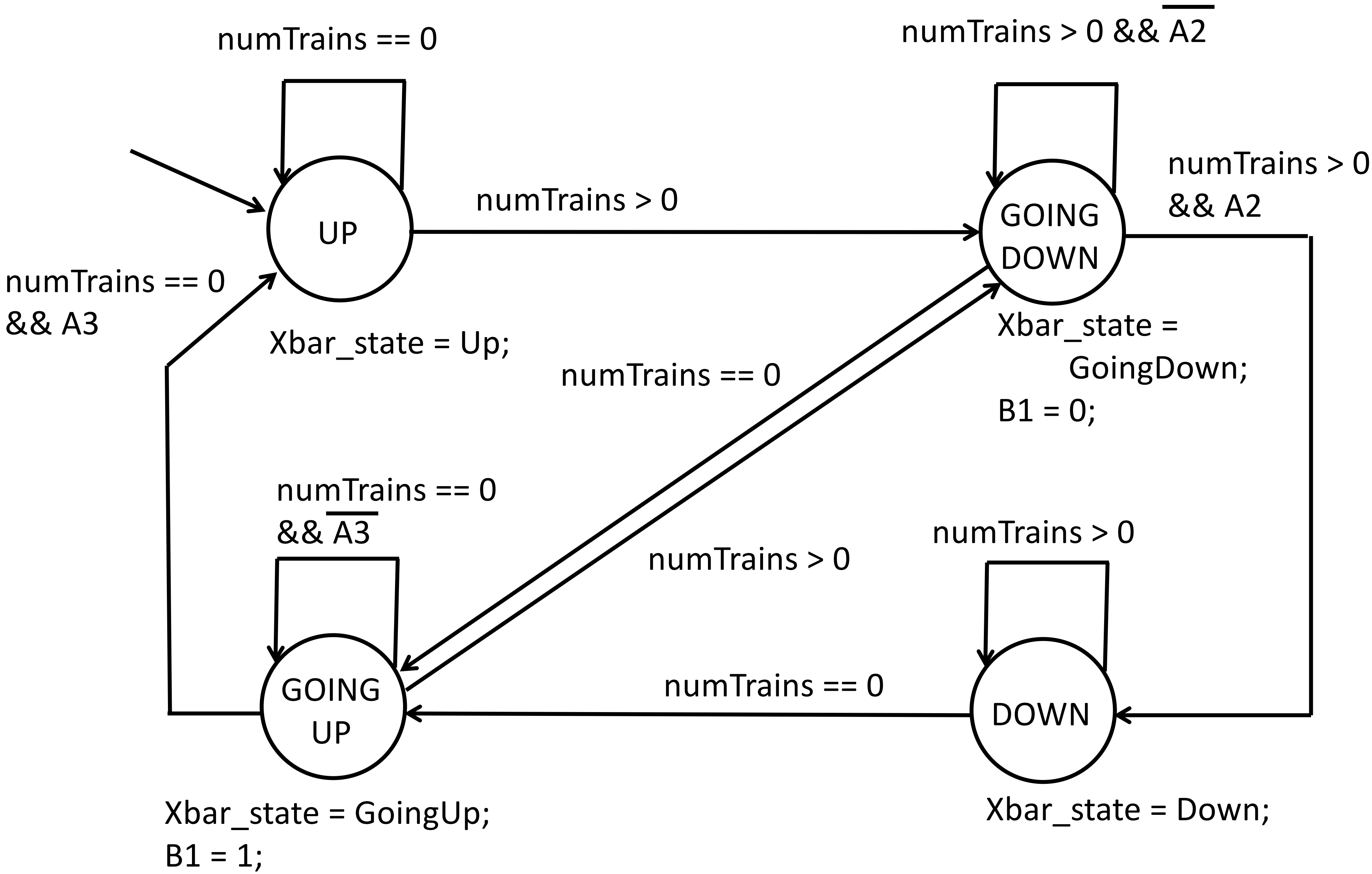
Task Diagram

Period = 100ms



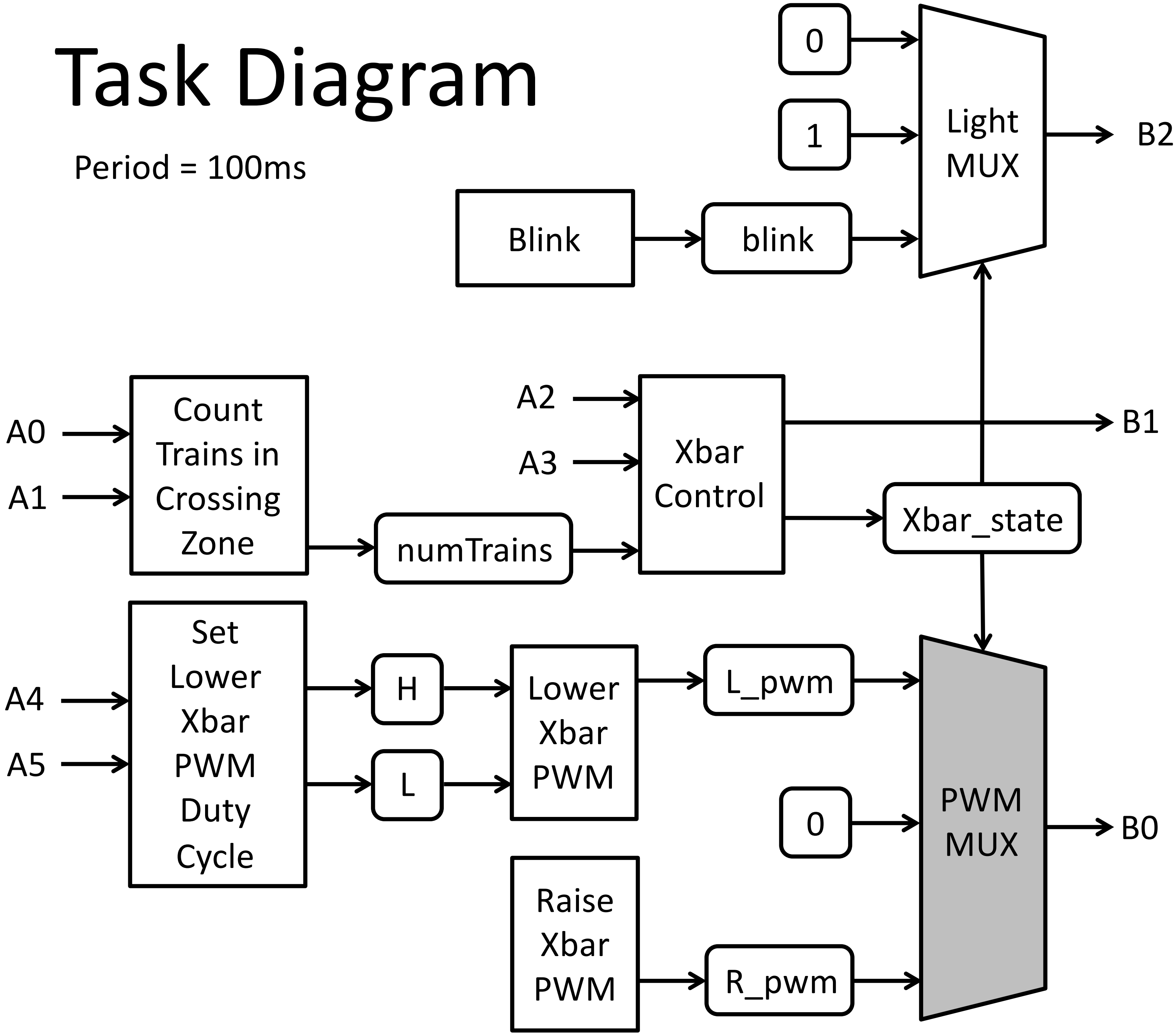
Xbar Control

Period = 100 ms

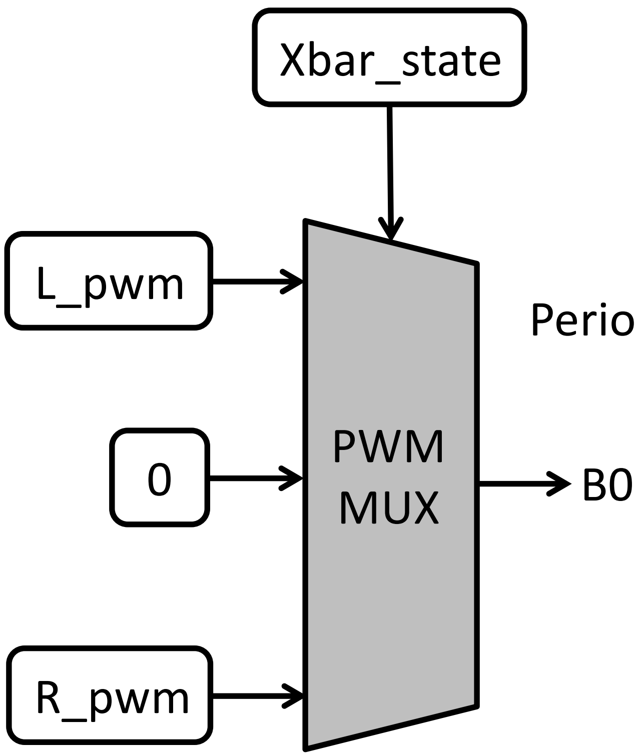


Task Diagram

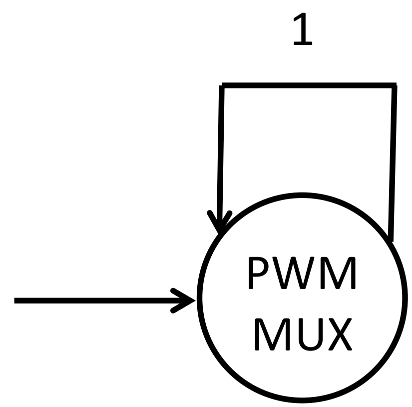
Period = 100ms



PWM Mux

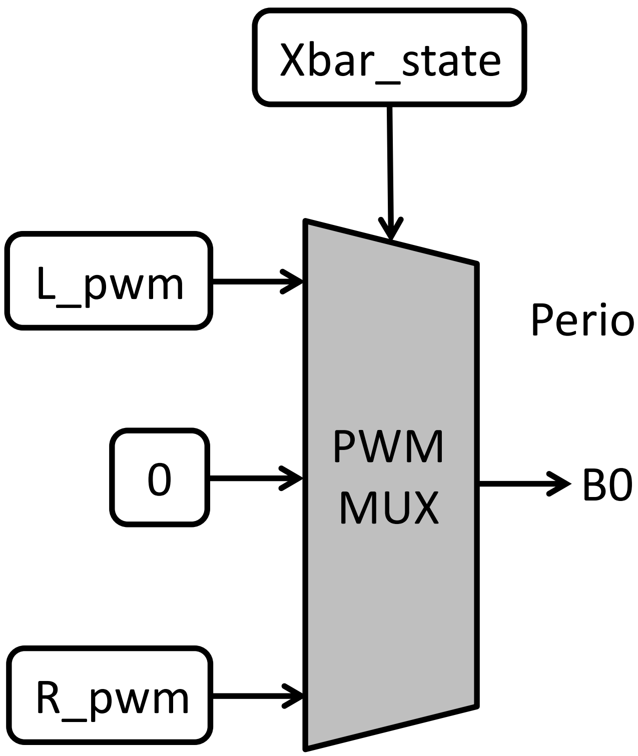


Period = 100 ms

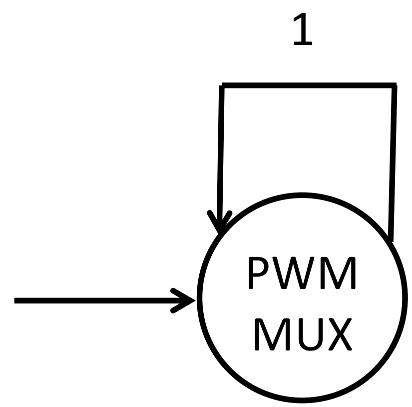



```
if( Xbar_state == GoingDown )
        B0 = L_pwm;
else if( Xbar_state == GoingUp )
        B0 = R_pwm;
else B0 = 0;
```

PWM Mux (Alternate)



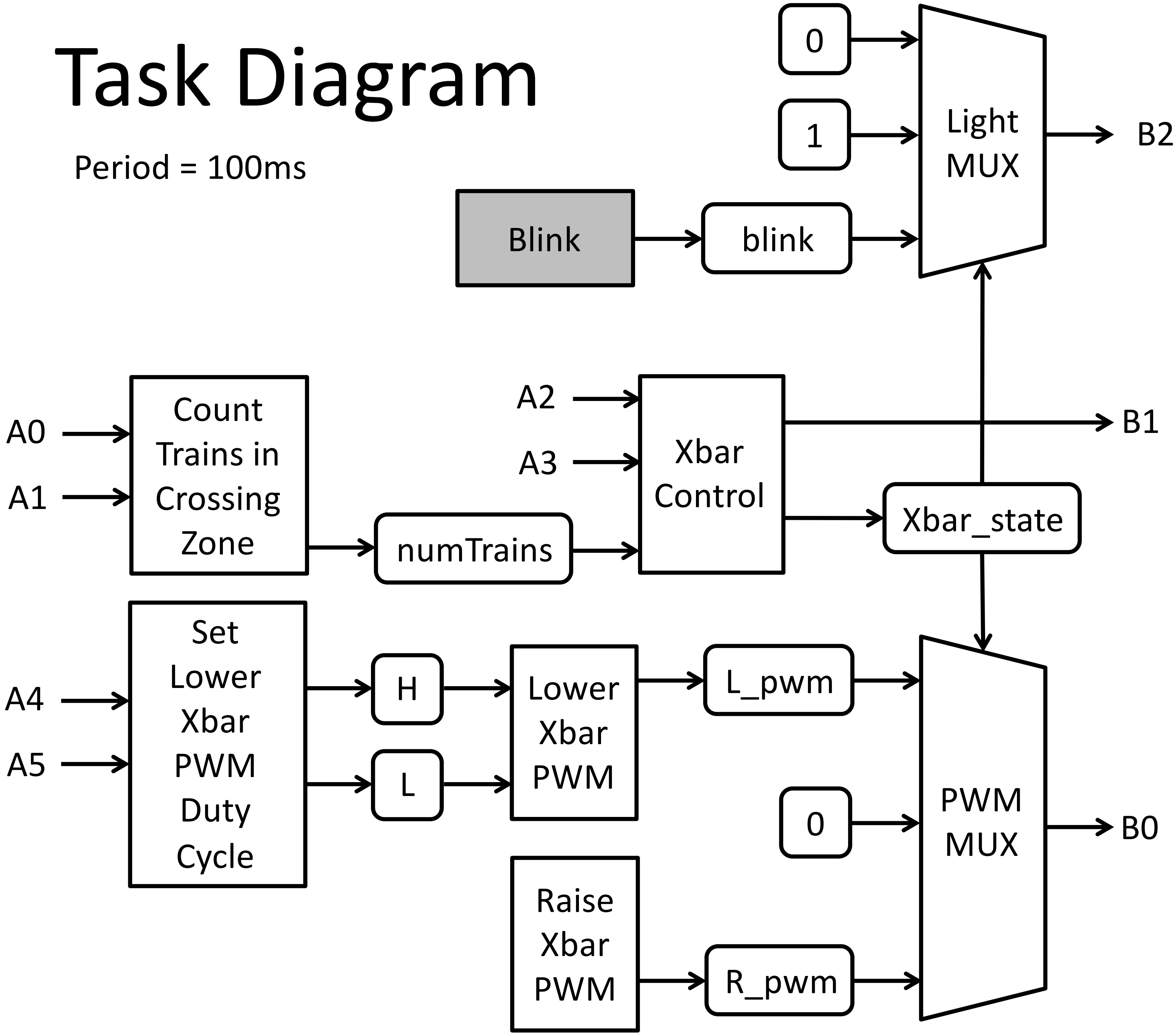
Period = 100 ms



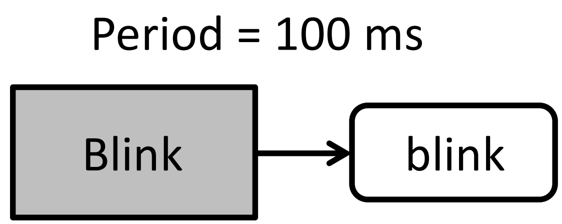

```
B0 = (Xbar_state == GoingDown)
        ? L_pwm
        : (Xbar_state == GoingUp)
                ? R_pwm
                : 0;
```

Task Diagram

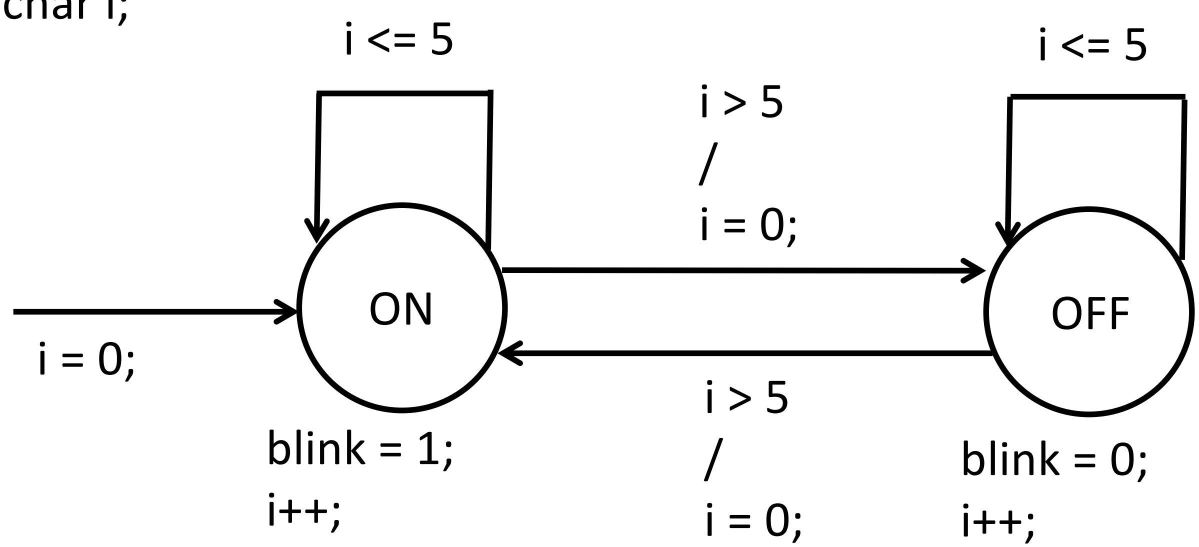
Period = 100ms



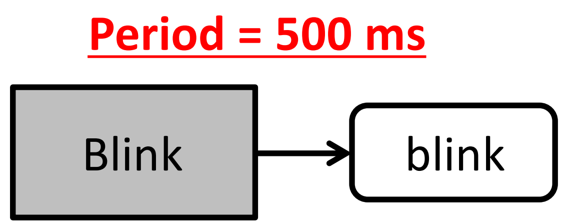
Blink

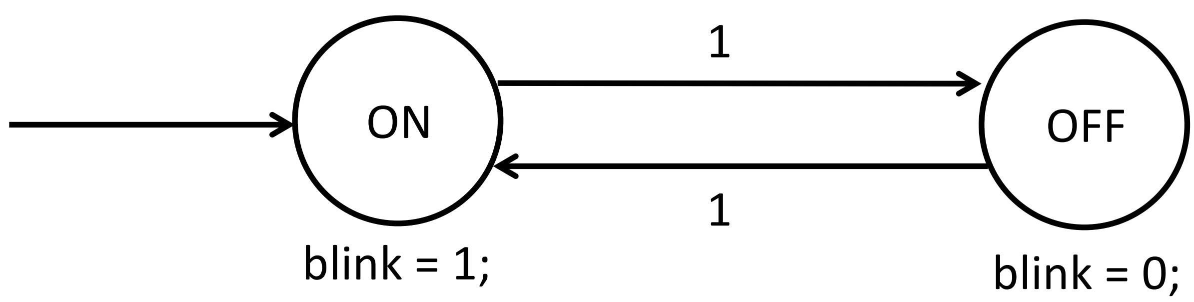


unsigned char i;



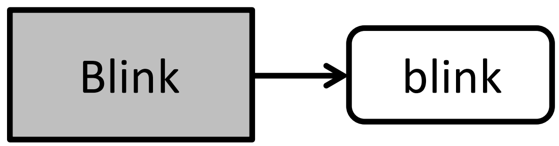
Blink (Simpler Version)



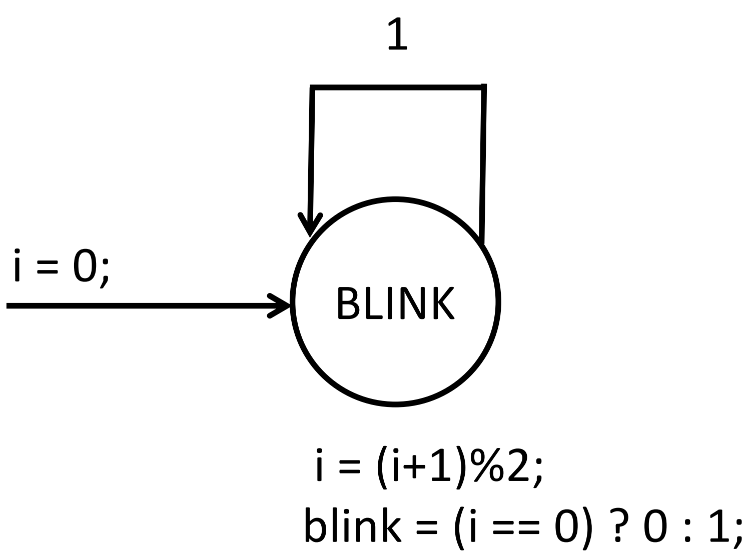


Blink (Simpler Version)

Period = 500 ms

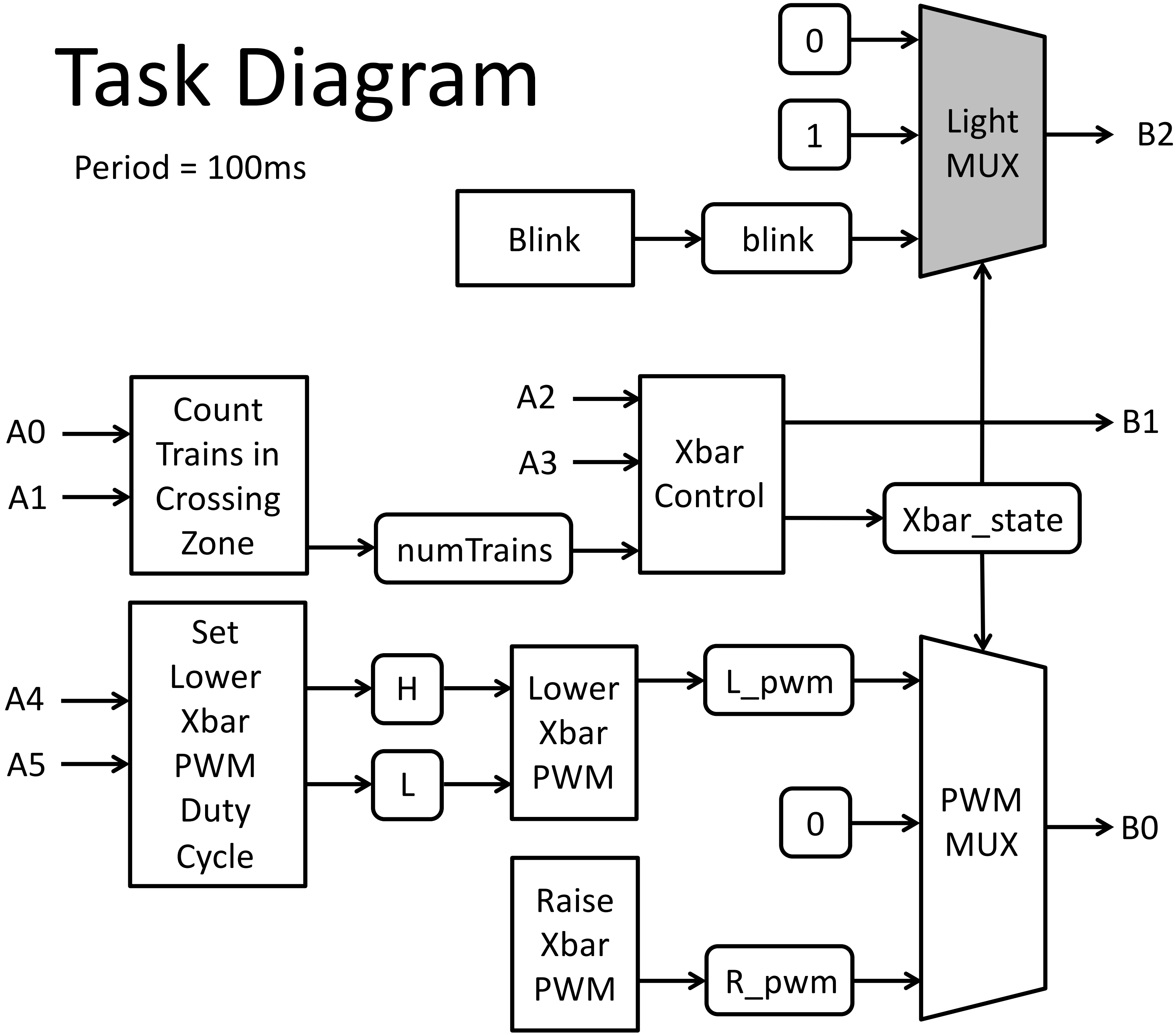


unsigned char i;



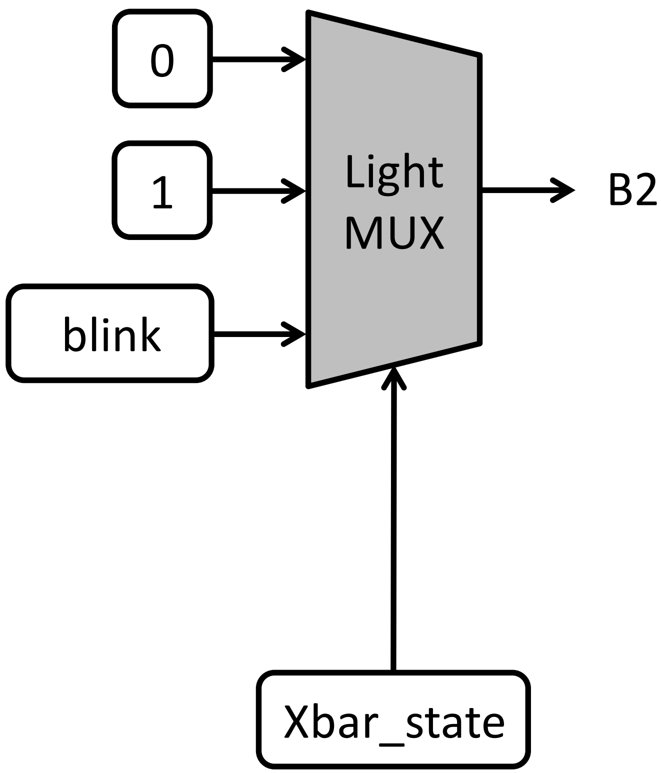
Task Diagram

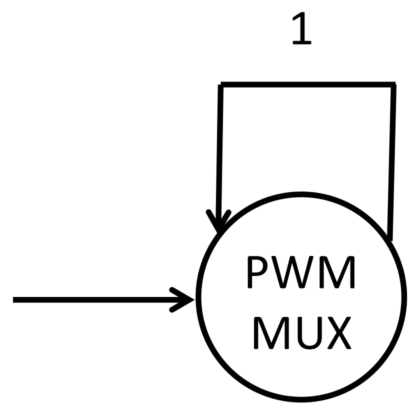
Period = 100ms



Light Mux

Period = 100 ms



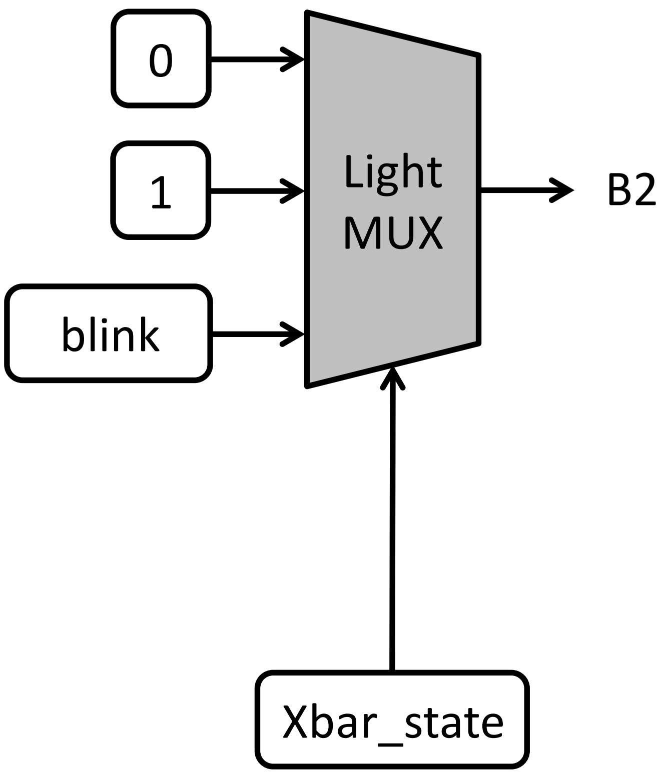


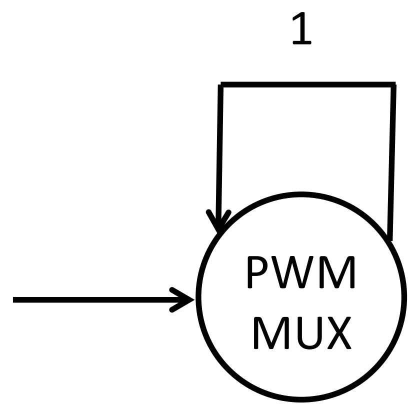
if(Xbar_state == Up) B2 = 0;
else if(Xbar_state == Down) B2 = 1;
else B2 = blink;

Light Mux (Alternate)

```
if( Xbar_state == Up ) B2 = 0;
else if( Xbar_state == Down ) B2 = 1;
else B2 = blink;
```

Period = 100 ms






```
B2 = ( Xbar_state == Up )
     ? 0
     : ( Xbar_state == Down )
          ? 1
          : blink;
```

Other Design Options

- 1- and 2-state implementations of the Xbar Control Task
- Combine the Blink and Light Mux tasks into a single task

- (Infinitely many others)