

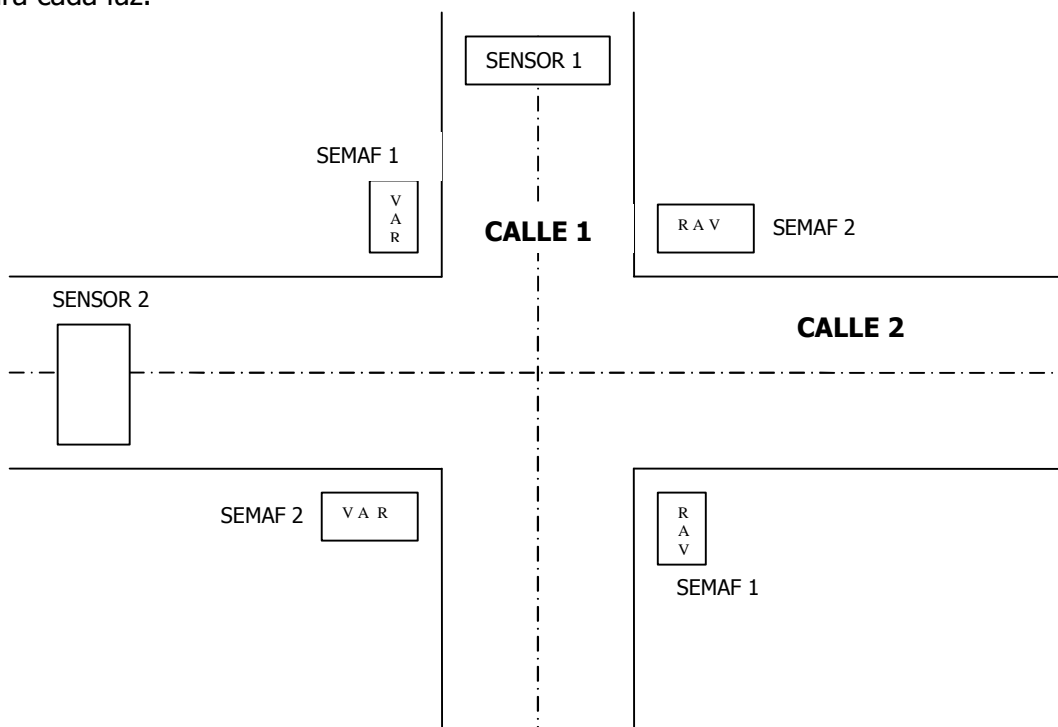
Ejemplo ABEL. Control de del tráfico con prioridad en un cruce.

Objetivo.

Se quiere hacer un controlador de cruce de semáforos. Este controlador tiene en cuenta la densidad del tráfico y modifica los tiempos de paso en cada calle dependiendo de la densidad del mismo. Las condiciones del tráfico se han simplificado para no hacer el problema demasiado complicado.

Como entradas se dispone de una señal de reloj y dos señales: S1=sensor1 indica si hay tráfico en la calle 1, y S2=sensor2 indica si hay tráfico en la calle 2.

Como salidas se tiene cada una de las luces de los semáforos (R1=rojo1, A1=ámbar1, V1=verde1, R2=rojo2, A2=ámbar2, V2=verde2). A pesar de haber dos semáforos, basta con una salida para cada luz.



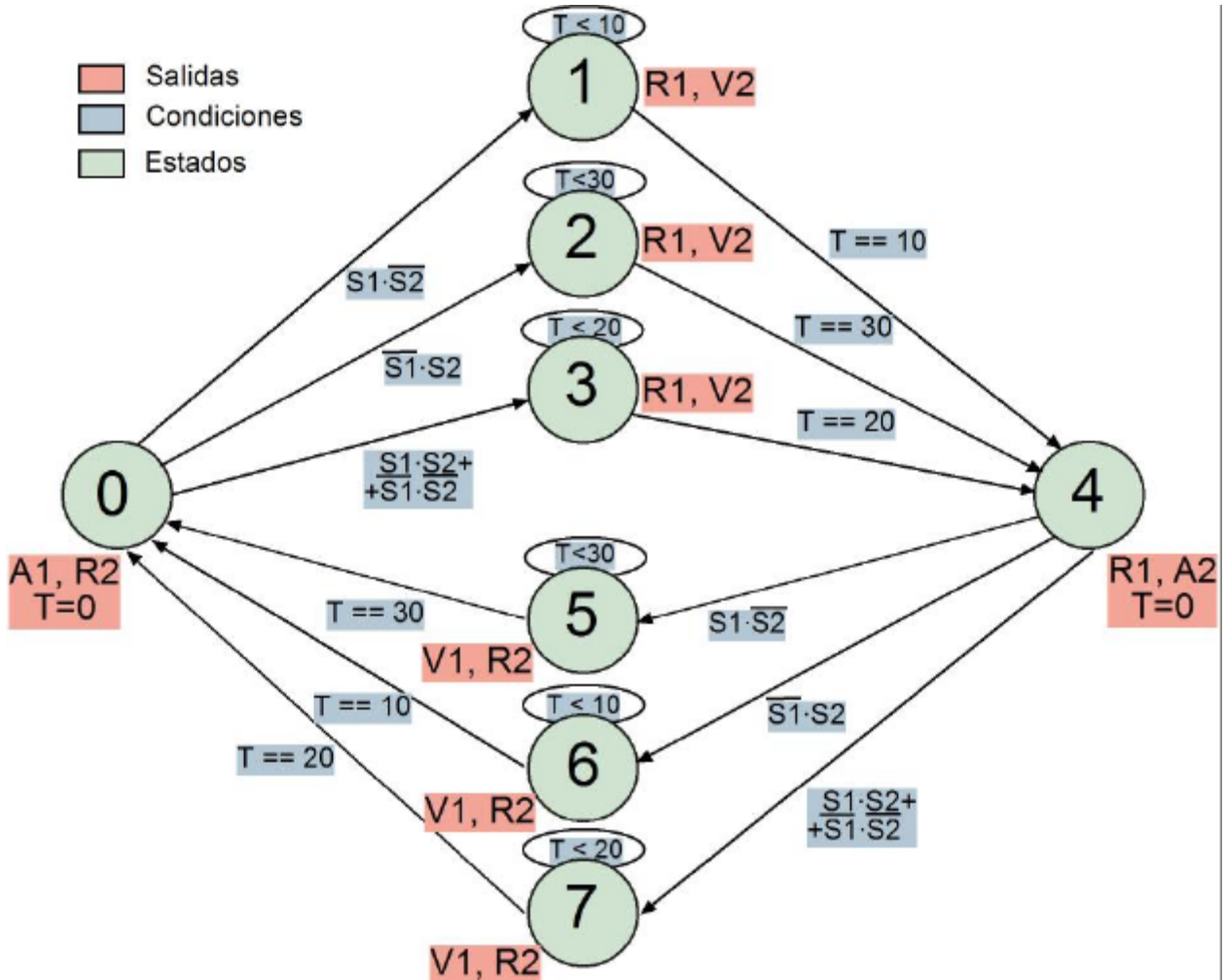
Descripción.

Si hay tráfico en ambas calles o no hay tráfico en ninguna de las dos calles, el tiempo de semáforo en verde para ambas calles es de $T=20$ segundos. Si hay tráfico en una calle y no lo hay en la otra, el tiempo de verde en la calle con tráfico es de $T=30$ segundos, mientras que el tiempo de rojo es de $T=10$ segundos.

Una vez que se ha evaluado la situación del tráfico y se han decidido los tiempos, éstos ya no se vuelven a modificar hasta que no ha finalizado el periodo de tiempo actual.

Resolución.

En este caso, dibujemos el diagrama de estados.



La decisión acerca del tiempo que debe permanecer en un estado se toma cuando el semáforo se encuentra en ámbar, a partir de ese momento el tiempo queda fijado hasta que se vuelve a producir un cambio a ámbar y se vuelven a evaluar las condiciones del tráfico en los dos sentidos de marcha.

Código ABEL.

El código ABEL sería:

```

module trafico
title 'modulo controlador de un semaforo de cruce con prioridad a la calle con mas trafico'

reloj, reset, sensor1, sensor2      PIN;
rojo1, ambar1, verde1              PIN istype 'com';
rojo2, ambar2, verde2              PIN istype 'com';
estado0, estado1, estado2          PIN istype 'reg';
tiempo4..tiempo0                   PIN istype 'reg';

estado = [estado2, estado1, estado0];
est0 = 0;      est1 = 1;      est2 = 2;
est3 = 3;      est4 = 4;      est5 = 5;
est6 = 6;      est7 = 7;

semaforo1 = [rojo1,ambar1,verde1];  semaforo2 = [rojo2,ambar2,verde2];
sensores = [sensor1, sensor2];      tiempo = [tiempo4..tiempo0];

equations

estado.clk = reloj;
estado.ar = reset;
tiempo.clk = reloj;
    
```

```

state_diagram estado
state est0:   semaforo1=[0,1,0];
              semaforo2=[1,0,0];
              tiempo.ar=1;
              if (sensor1 & !sensor2) then est1;
              else if (!sensor1 & sensor2) then est2;
              else est3;
state est1:   semaforo1=[1,0,0];
              semaforo2=[0,0,1];
              tiempo := tiempo + 1;
              if tiempo == 10 then est4 else est1;
state est2:   semaforo1=[1,0,0];
              semaforo2=[0,0,1];
              tiempo := tiempo + 1;
              if tiempo == 30 then est4 else est2;
state est3:   semaforo1=[1,0,0];
              semaforo2=[0,0,1];
              tiempo := tiempo + 1;
              if tiempo == 20 then est4 else est3;
state est4:   semaforo1=[1,0,0];
              semaforo2=[0,1,0];
              tiempo.ar=1;
              if (sensor1 & !sensor2) then est5;
              else if (!sensor1 & sensor2) then est6;
              else est7;
state est5:   semaforo1=[0,0,1];
              semaforo2=[1,0,0];
              tiempo := tiempo + 1;
              if tiempo == 30 then est0 else est5;
state est6:   semaforo1=[0,0,1];
              semaforo2=[1,0,0];
              tiempo := tiempo + 1;
              if tiempo == 10 then est0 else est6;
state est7:   semaforo1=[0,0,1];
              semaforo2=[1,0,0];
              tiempo := tiempo + 1;
              if tiempo == 20 then est0 else est7;
end trafico

```