

8.1 código c

/*César Augusto B. Gonçalves

Escreva um simples programa em C no RIMS que seta B com A exceto que B1 e B0 são sempre 0. Use as duas formas, bitwise e booleana. Teste após a escrita*/

```
#include "rims.h"
```

```
void main()
```

```
{
```

```
    while (1) {
```

```
        //B = (A << 2);
```

```
        B = (A & 0xFC);
```

```
    }
```

```
}
```

8.2 código c

/*César Augusto B. Gonçalves

Considere um sistema com um sensor de som com uma saída de 4-bits conectada a A3...A0, o (0000) significa sem som, 15 (1111) significa som alto. Um segundo sensor conecta A7...A4. Um sistema deve adicionar esses dois valores de 4-bits (soma) na saída B */

```
#include "rims.h"
```

```
void main()
```

```
{
```

```
    unsigned char som1 = 0;
```

```
    unsigned char som2 = 0;
```

```
    while (1) {
```

```
        som1 = A & 0x0f;
```

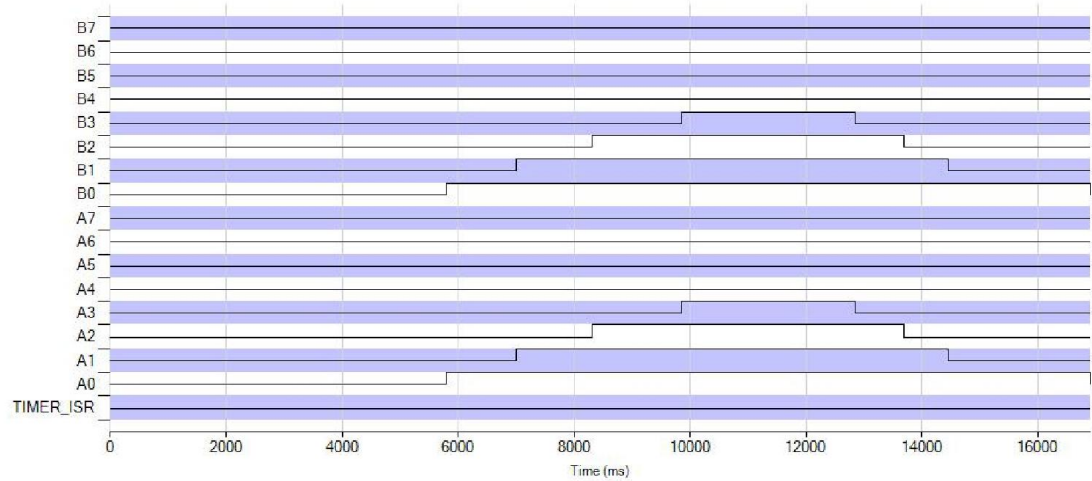
```
        som2 = A >> 4;
```

```
        B = som1 + som2;
```

```
    }
```

```
}
```

8.2 timing_diagram



8.2 vetor_de_testes

```
b000000000
wait 1 s
3
wait 1 s
assert 1
assert b000000001
assert 0x01
wait 0.5 s
b000000010
wait 3000 ms
b111111101
wait 200 ms
b00010101
wait 500 ms
b110000011
wait 2 s
b010000010
wait 1 s
0xFF
wait 1 s
generatetd
```



8.3 código c

/*César Augusto B. Gonçalves

Escreva um programa no RIMS que seta B3-B0 para A5-A2 e seta as demais saídas com 0s. Teste no RIMS e crie um vetor de teste.

*/

```
#include "rims.h"
```

```
void main()
```

```
{  
    while (1) {  
        B = (A >> 2) & 0x0F;  
    }  
}
```

8.3 vetor_de_testes

```
b000000000  
wait 1 s  
3  
wait 1 s  
assert 1  
assert b00000001  
assert 0x01  
wait 0.5 s  
b000000010  
wait 3000 ms  
b111111101  
wait 200 ms  
b00010101  
wait 500 ms  
b11000011  
wait 2 s  
b01000010  
wait 1 s  
0xFF  
wait 1 s  
generatetd
```

8.4 código c

/*César Augusto B. Gonçalves

[Sensor de vaga de estacionamento] Um estacionamento tem 8 espaços, cada um com um sensor conectada a entrada do RIMS (A7...A0). Um entrada sendo 1, significa que um carro foi detectado no espaço correspondente. A7 e A6 são reservados para deficientes. Escreva um programa no RIMS que: (1) Seta B0 com 1 se ambos os espaços de deficientes estão ocupados;

(2) Seta B7...B5 igual ao número de espaços disponíveis para não deficientes */

```
#include "rims.h"
```

```
void main()
```

```
{
```

```
    while (1) {
```

```
        B = (A6 && A7) + ((A0 + A1 + A2 + A3 + A4 + A5) << 5);
```

```
    }
```

```
}
```

8.4 vetor_de_testes

```
b000000000
```

```
wait 1 s
```

```
3
```

```
wait 1 s
```

```
assert 1
```

```
assert b000000001
```

```
assert 0x01
```

```
wait 0.5 s
```

```
b000000010
```

```
wait 3000 ms
```

```
b111111101
```

```
wait 200 ms
```

```
b00010101
```

```
wait 500 ms
```

```
b11000011
```

```
wait 2 s
```

```
b01000010
```

```
wait 1 s
```

```
0xFF
```

```
wait 1 s
```

```
generatetd
```