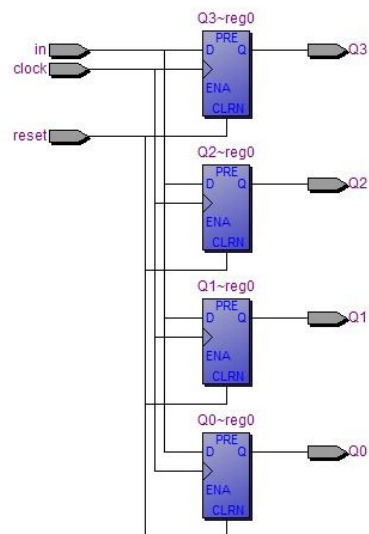
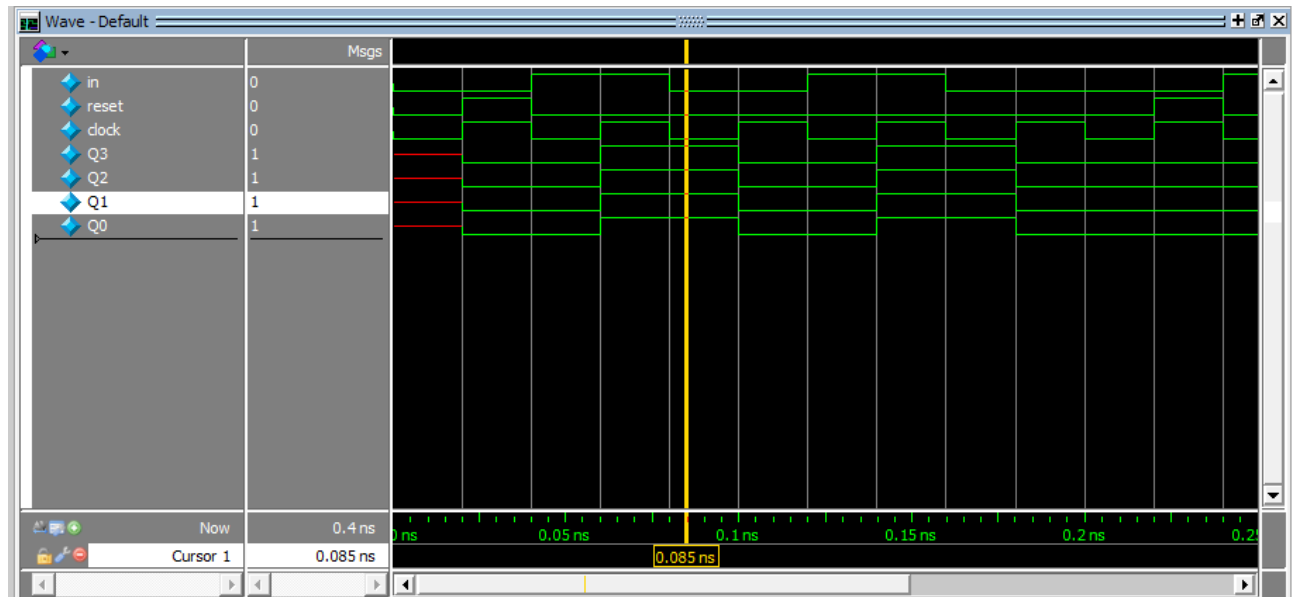


Nome: Ryan Eduardo Mansur Vasconcelos e Anélio Gonçalves Caldas

## Registrador usando atribuição bloqueante.



### Variáveis:

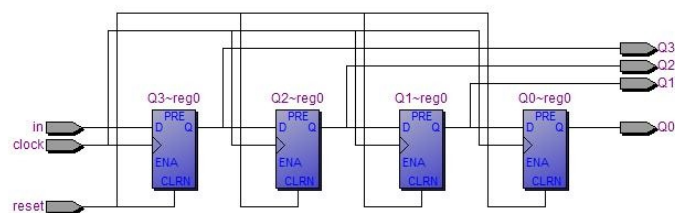
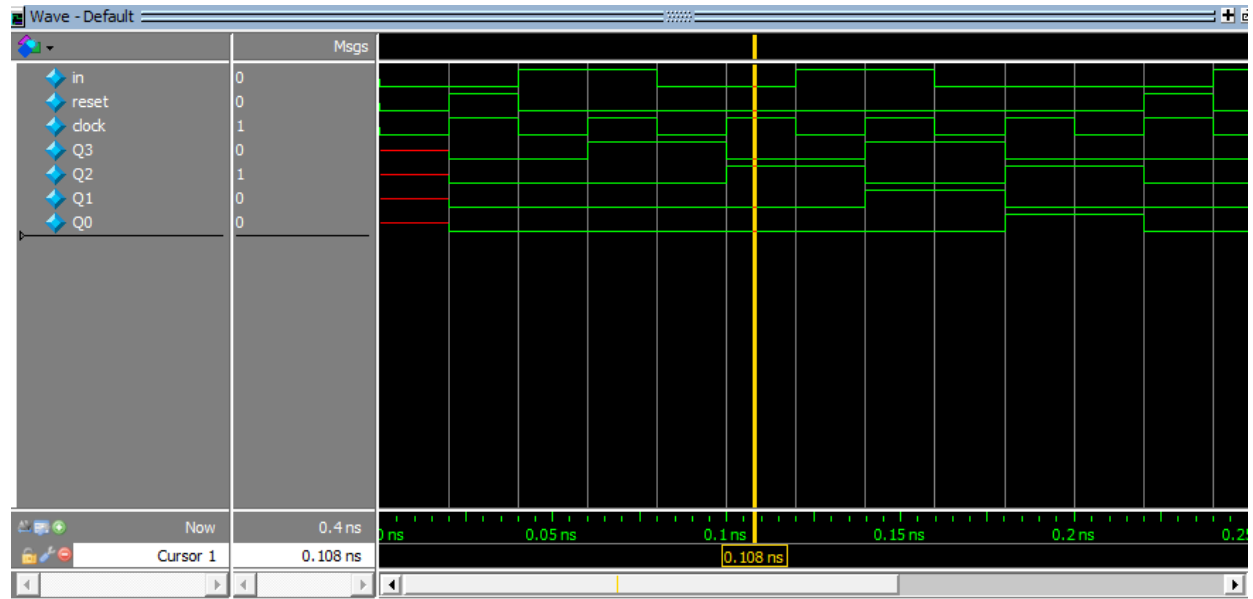
In: A variável de entrada.

Clock: Variável com detecção de borda de subida

Reset: Zera todas as saídas.

Q: As saídas.

## Registrador sem usar atribuição bloqueante



### Variáveis:

In: A variável de entrada.

Clock: Variável com detecção de borda de subida

Reset: Zera todas as saídas.

Q: As saídas.

### **Considerações finais.**

O Registrador, usando atribuição bloqueante não funciona, a partir do momento que se usa o Q anterior para setar o próximo Q, todos irão pegar o mesmo bit, e assim, não deslocando o bit, e sim trocando todos pelo número setado no in.

No outro caso, não temos esse problema, a atribuição feita em conjunta favorece esse deslocamento, fazer com que o registrador funcione como o esperado.

Na simulação, foi visto a funcionalidade do registrador e da borda de subida, foi setado pelo testbench, que a simulação reiniciasse apenas 2 vezes, assim podendo fazer a testagem da função do Reset, também foi usado o in, de forma como ele mudasse sem a mudança no clock, e como se espera, quando o clock não mudasse de 0 a 1, mesmo o In mudando, não havia deslocamento.

No geral, a simulação ocorreu como o esperado quando se usa atribuição não bloqueante, o erro no outro caso já era visível na imagem que o Quartus gera.