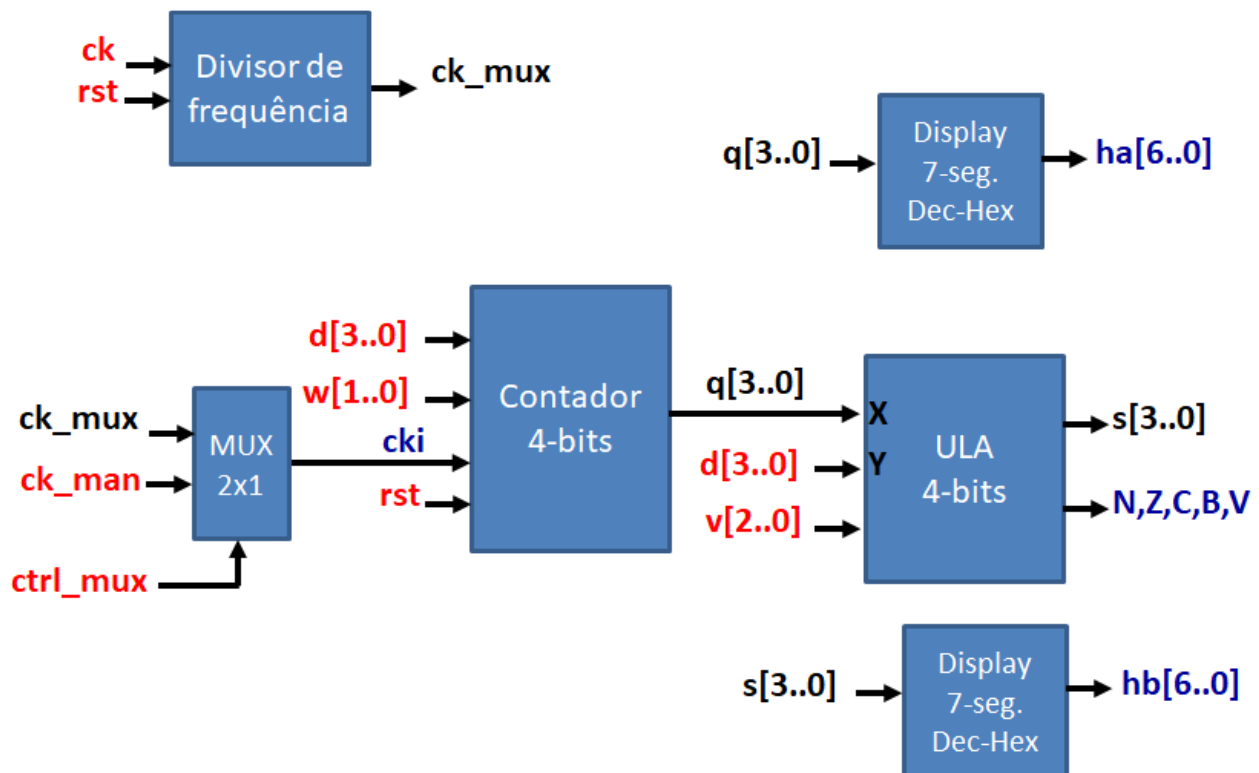


### Projeto Final



### Para colocar na placa de prototipagem

#### Entradas:

- ck: clock interno (disponível) da placa;
- rst: botão de apertar (*push botton*);
- ck\_man: botão de apertar (*push botton*);
- switches (10): d[3..0], v[2..0], w[1..0], ctrl\_mux

#### Saídas:

- cki: ponto decimal do display de q[3..0];
- ha[6..0], hb[6..0]: display 7-degmentos);
- N,Z,C,B,V: leds verdes;

### Especificação do Contador

Projeto de um **contador síncrono** de 4 bits com 4 configurações de operação:

W1	W0	Operação
0	0	Mantém a saída inalterada ( <i>keep</i> )
0	1	Incrementa o contador +1 ( <i>up counter</i> )
1	0	Decrementa o contador -1 ( <i>down counter</i> )
1	1	Carrega a entrada D (4 bits) do contador ( <i>load</i> )

## Especificação da ULA

Projeto de uma unidade lógica e aritmética (**ULA**) de 4 bits com 8 operações conforme a tabela abaixo:

V2	V1	V0	Operação
0	0	0	Passa X ( $S = X$ )
0	0	1	X AND Y
0	1	0	X OR Y
0	1	1	NOT (X) <i>negação em complemento de 1</i>
1	0	0	NEG (X) <i>negação em complemento de 2</i>
1	0	1	(ADD) $X + Y$ (soma em complemento de 2)
1	1	0	(SUB) $X - Y$ (subtração em complemento de 2)
1	1	1	(SHL X) Deslocamento à esquerda (x2) - <i>shift left</i> (X)

\* X e Y são vetores de 4 bits, assim como a saída S.

Devem ser gerados os seguintes sinais de condição (flags):

**Z (zero), N (negativo), C (carry), B (borrow) e V (overflow).**

Obs.: As operações NEG e SUB, em complemento de 2, podem ser construídos a partir de um somador:  $X - Y = X + \text{NOT}(Y) + 1$ . Portanto, é possível ter apenas um somador de 4 bits na ULA para realizar as 3 operações. Essa otimização será considerada na avaliação do trabalho.

### Entrega:

- Entregar, em arquivo único compactado, todos os arquivos BDF e BSF envolvidos (não inclua outros arquivos).