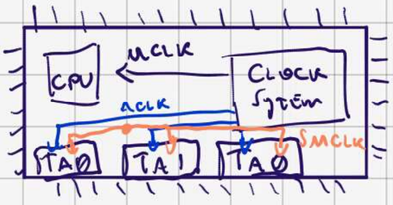
* temporizadores (Timers)

- → contadores
- → referência de tempo
- → comparadores



⇒ Referência de tempo:

- MCLK: master clock: 16MHZ
- ACLK: Auxiliar clock: 32768 HZ
- SMCLK: Submodule clock: 1MHZ

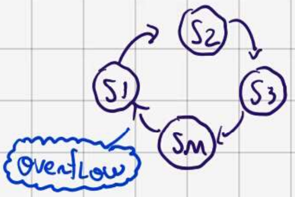
↳ Baixar o template setclock.zip

- pmm.h ←
- pmm.c
- clock.h ←
- clock.c
- main.c

```
SetVcoreUp(1);
SetVcoreUp(2);
Clock Init();
...
CODIGO
...
```

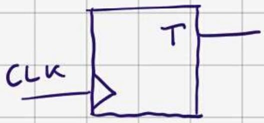
⇒ Contadores:

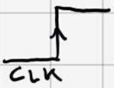
máquinas de estados cíclicos



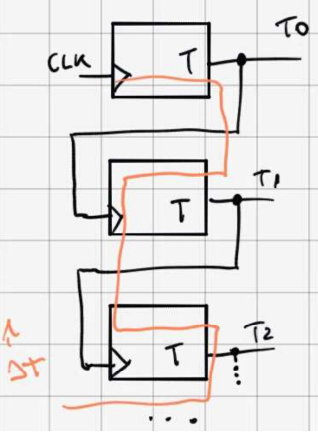
contador de módulo M.

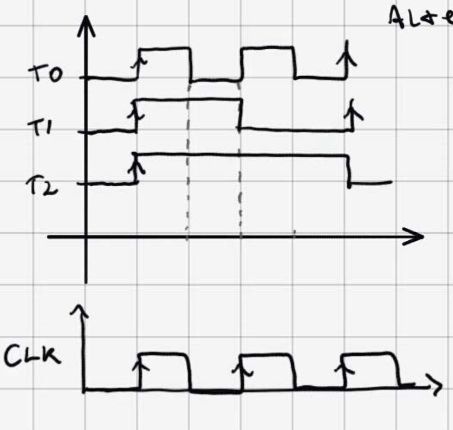
→ flip-flop toggle

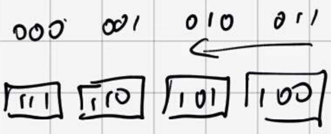




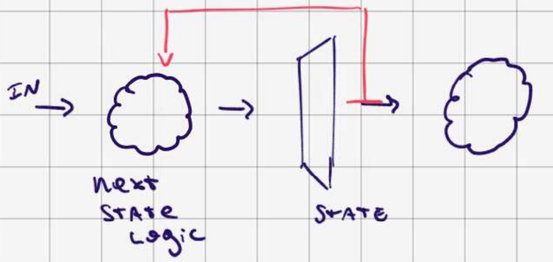
→ T → $\overline{T}$ "A cada flanco de CLK T alterna o valor"

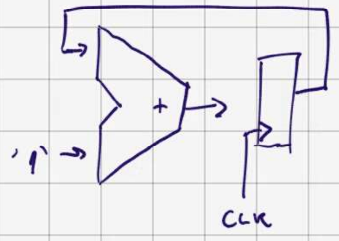


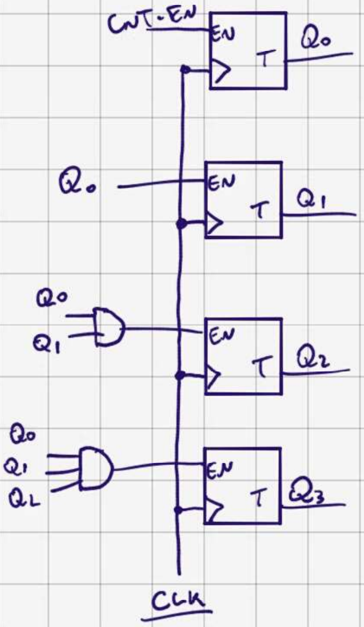




Design em FSM:





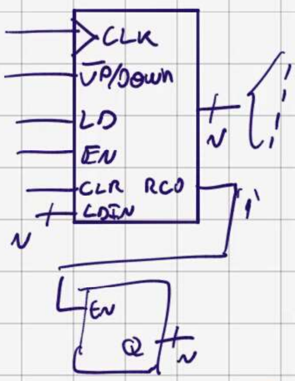


Contagem binária

000
001
010
011
100
101
110
111
1000

CONTADORES COMERCIAIS

74163, 74169



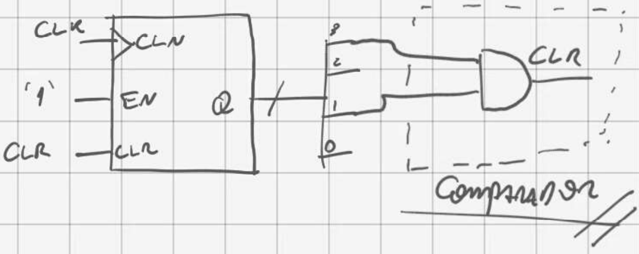
Ex: USANDO CONTADORES COMERCIAIS, crie um contador de [0-10].

→ Como é representado $(10)_{10}$ em binário?

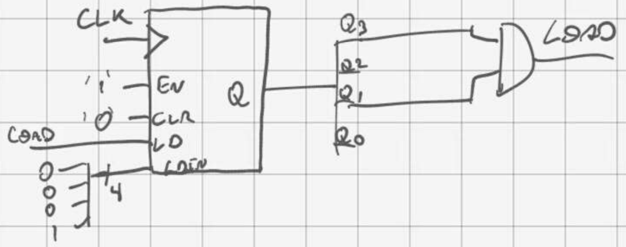
→ Como usar essa condição (Q=10) para reiniciar a contagem?

$Q_3 Q_2 Q_1 Q_0 = (10)_{10} = (1010)_2$

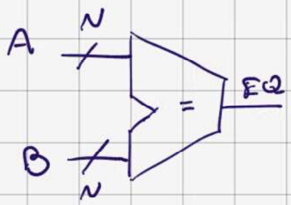
$(Q_3 \text{ e } Q_1) = '1'$



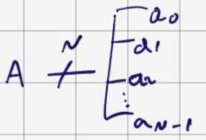
Ex 2: Contar [1-10]



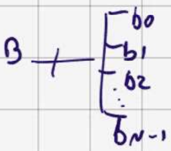
→ Comparador de uso geral:

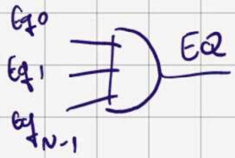


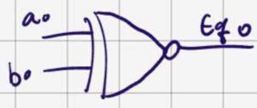
EQ = 1 quando A = B

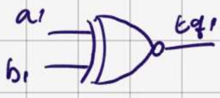


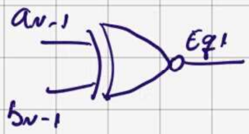
a	b	xor	xnor
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1



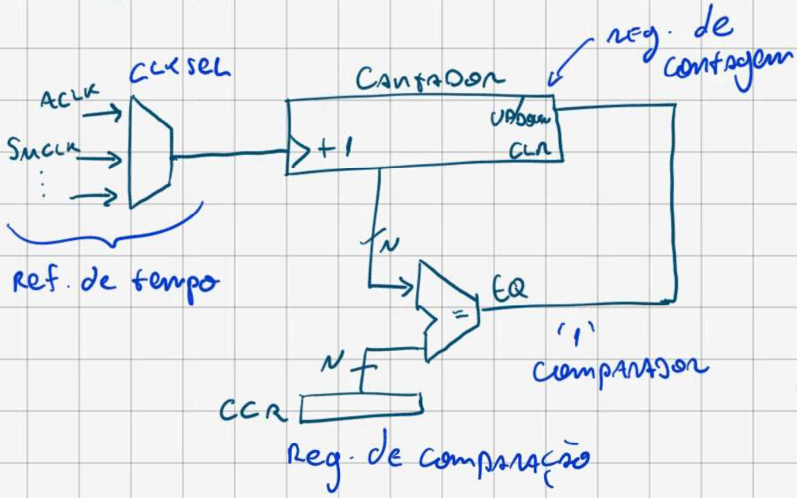


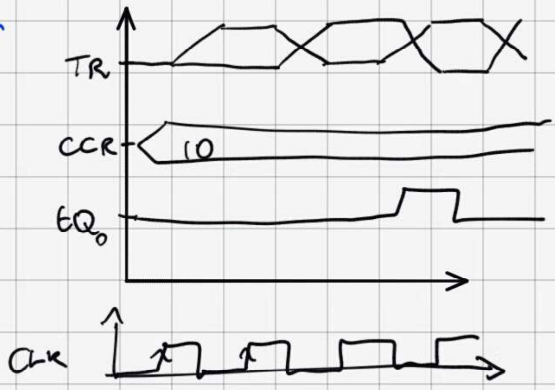




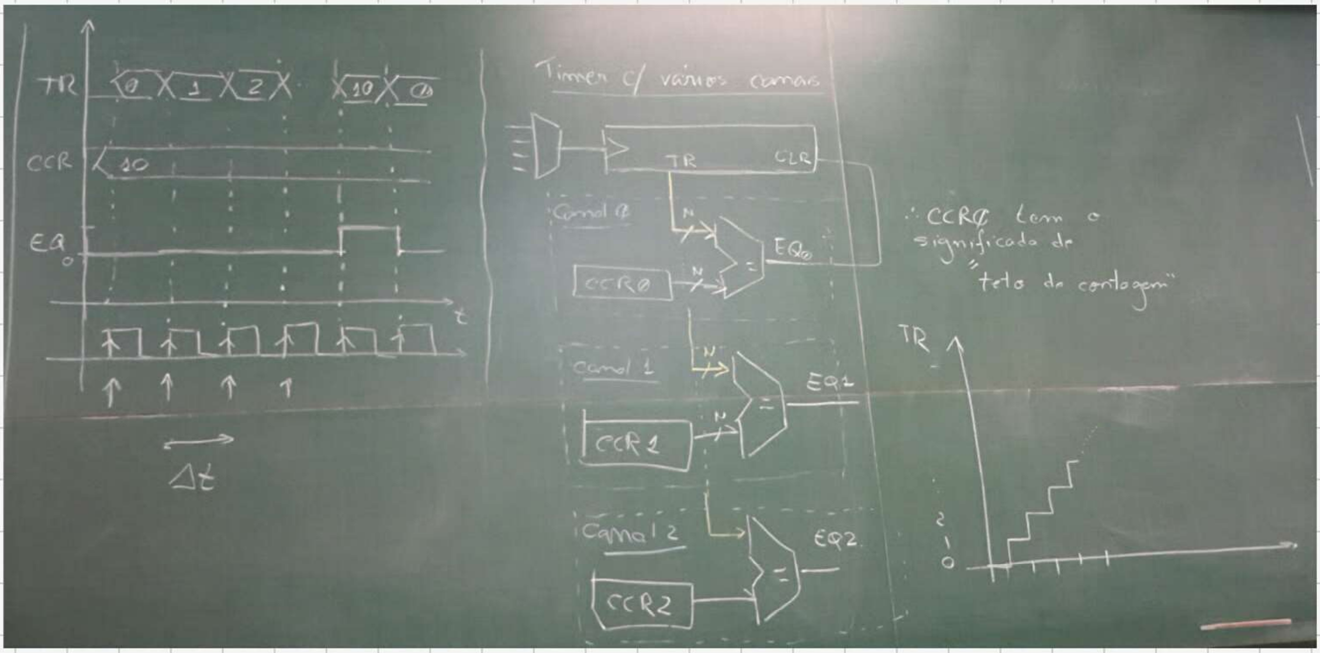


<Estrutura>

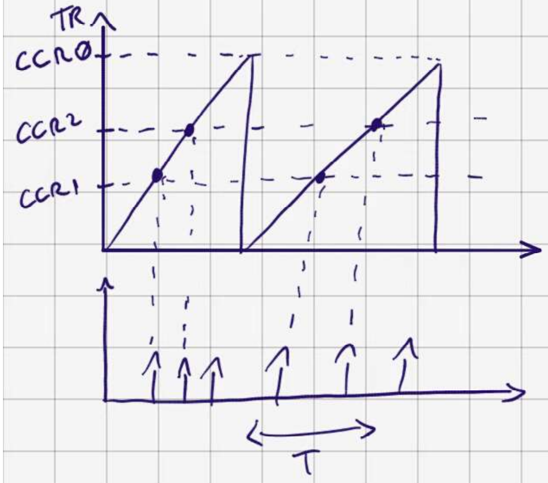




timer c/ vários canais:



∴ CCR0 tem o significado de "teto de contagem"



"Canal Ø tem uma ISR só p/ele"

"Canais 1 → N estão agrupados numa ISR"

* Config. específica do MSP430

TAØ, TA1, TA2

↳ contador: TAØR
TAØCTL

Comparador: TAØCCRØ
TAØCCTLØ — Canal 0

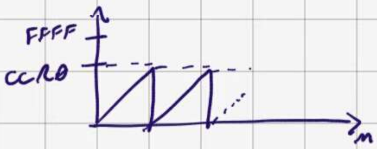
TAØCCR1
TAØCCTL1 — Canal 1

TAØCCR2
TAØCCTL2 — Canal 2

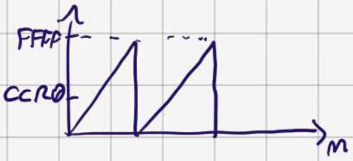
TAxCCRy
TAxCCTLy
↑
bitØ → IFG

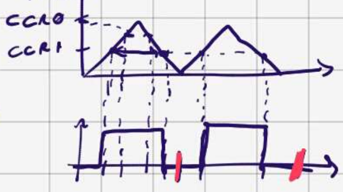
*Modos de contagem:

MC__HOLD → Ø
MC__UP → 1



MC__CONTINOUS → 2





MC__UPDOWN → 3

configurar o timer : [0 - 10]

```
  configtimer(){
 //configuração
      TA0CTL = TASSEL__ACLK | MC__UP | TACLR;

//Comparadores

      TA0CCR0 = 10;
  }


 main(){

     configtimer();

     While(1){
           while(!(TA0CCTL0 & TAIFG));

           P1OUT ^= BIT0;

           TA0CCTL0 &= ~TAIFG;
     }
```