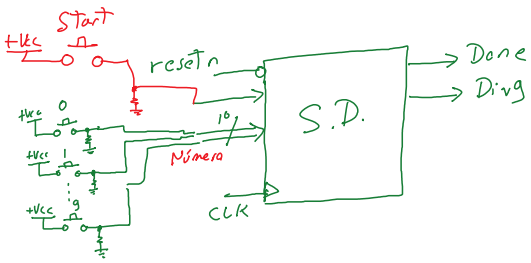
steps

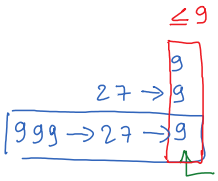
1. Start
2. Número ("9", "8", "1")
3. Sumar dígitos 9x999 = 8991 → 27 → 9
4. Done y Divg
5. Start → inicio

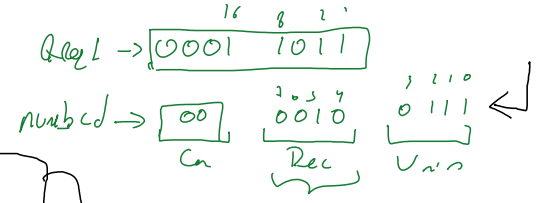


Pseudocódigo.-

```
while(1) {
  while(sTart==0) { }
  while(OR(Número)==0) { }
  RegCen = Decod_Decimal_BCD(Numero);
  while(OR(Número)==0) { }
  RegDec = Decod_Decimal_BCD(Numero);
  while(OR(Número)==0) { }
  RegUni = Decod_Decimal_BCD(Numero);
}
```

≤ 9





Desafío #1

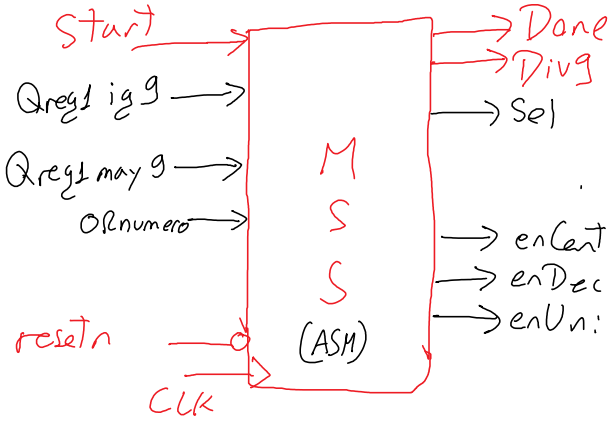
```
Load:
  QReg1 = Add4("0", RegCen, RegDec, RegUni);
  num_bcd = Decobin_BCD(QReg1);
  QReg2 = Add(num_bcd(3..0), num_bcd(7..4));
Result:
  while(start==0) {
    Done = 1;
    if(QReg2==9){
      Divg = 1;
    }
  }
```

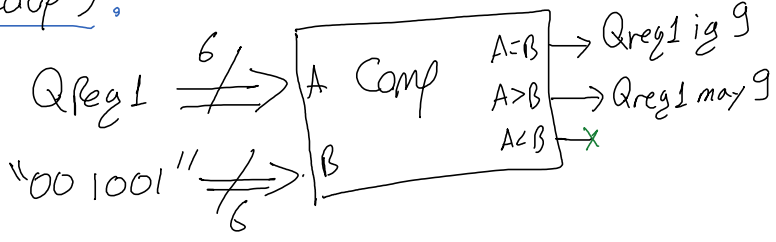
Loop

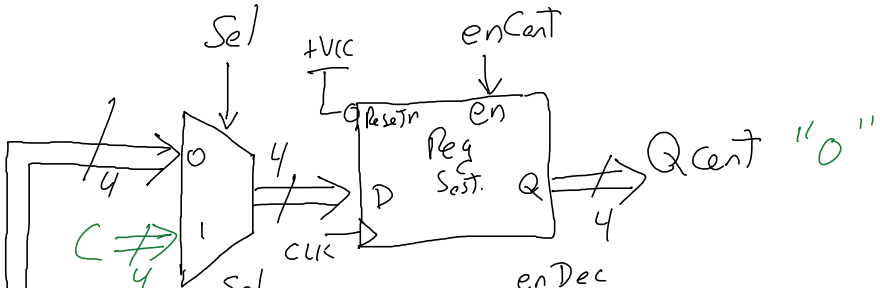
```
while(QReg1 > 9){
  QReg1 = Add4("0", RegCen, RegDec, RegUni);
  num_BCD = Decobin_BCD(QReg1);
  RegCen = num_BCD(10..8);
  RegDec = num_BCD(7..4);
  RegUni = num_BCD(3..0);
}
```

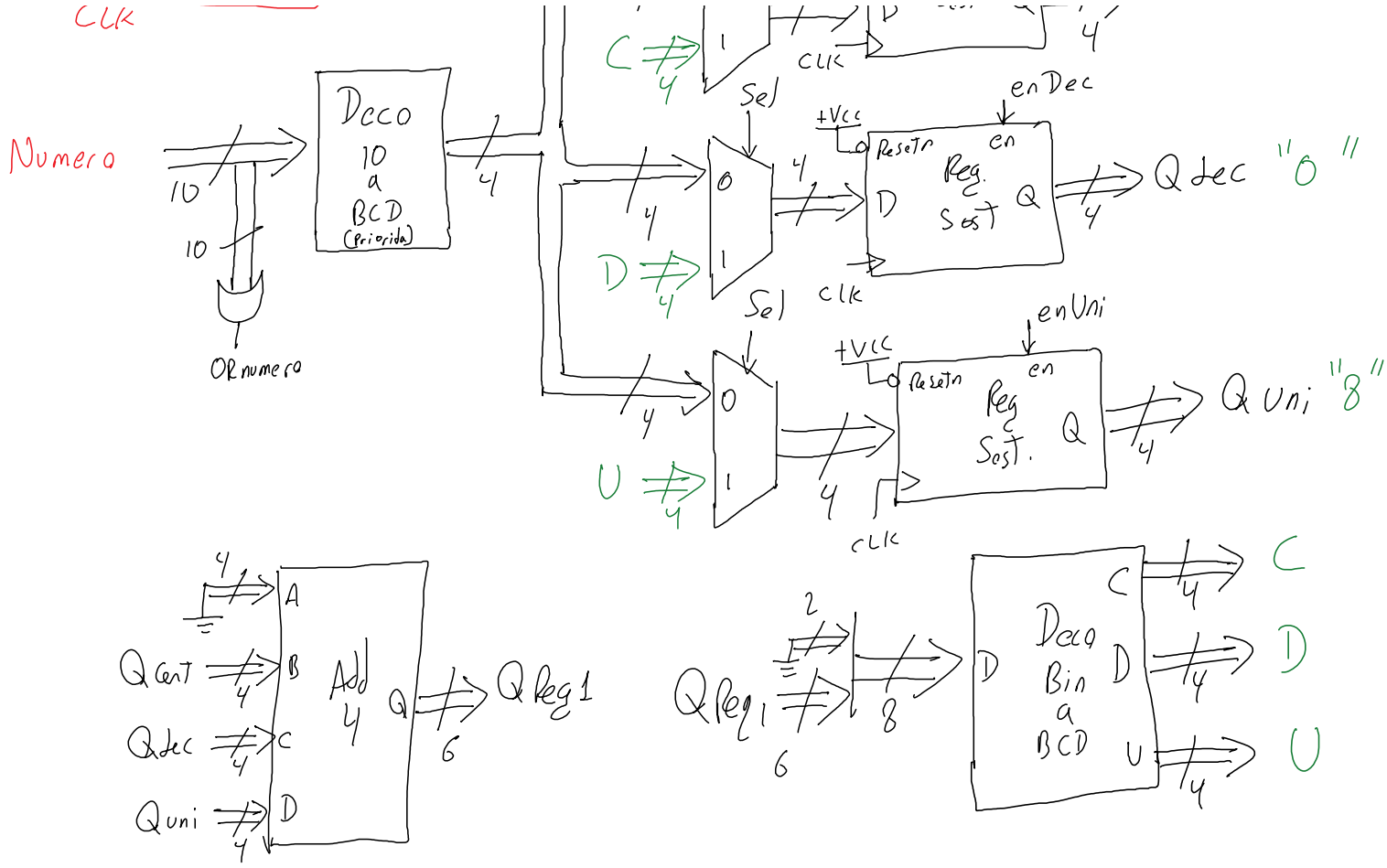
6bits

Partición Funcional (Versión Loop).-

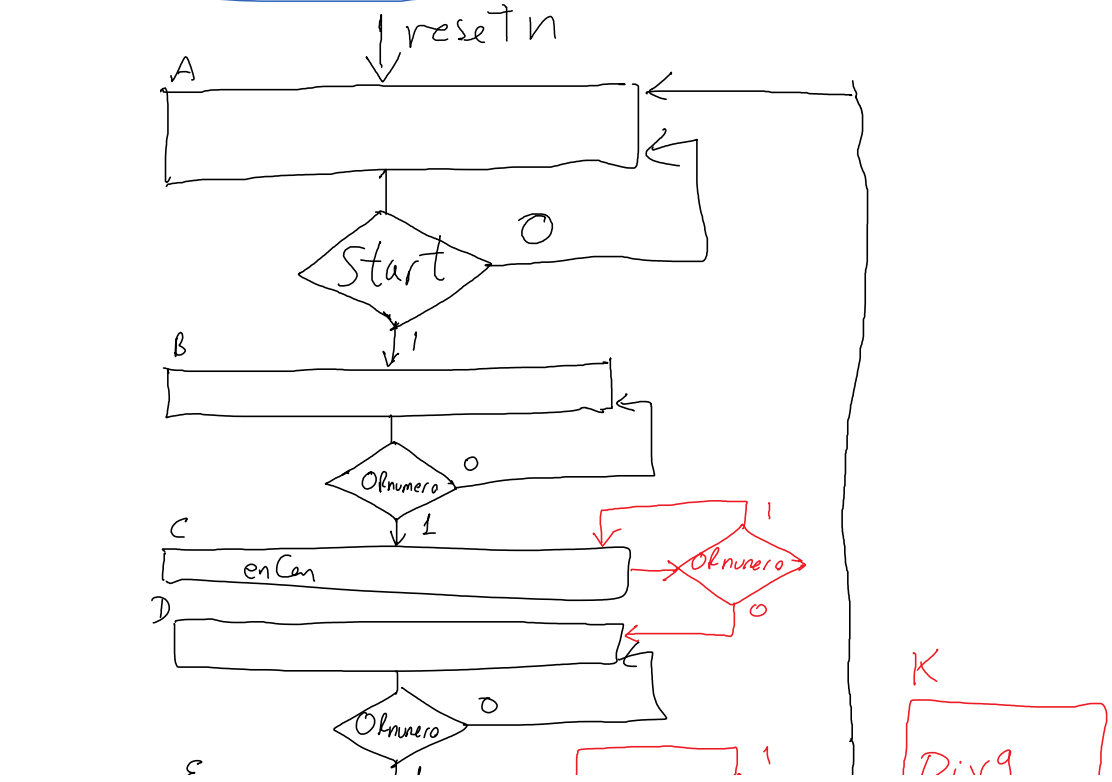


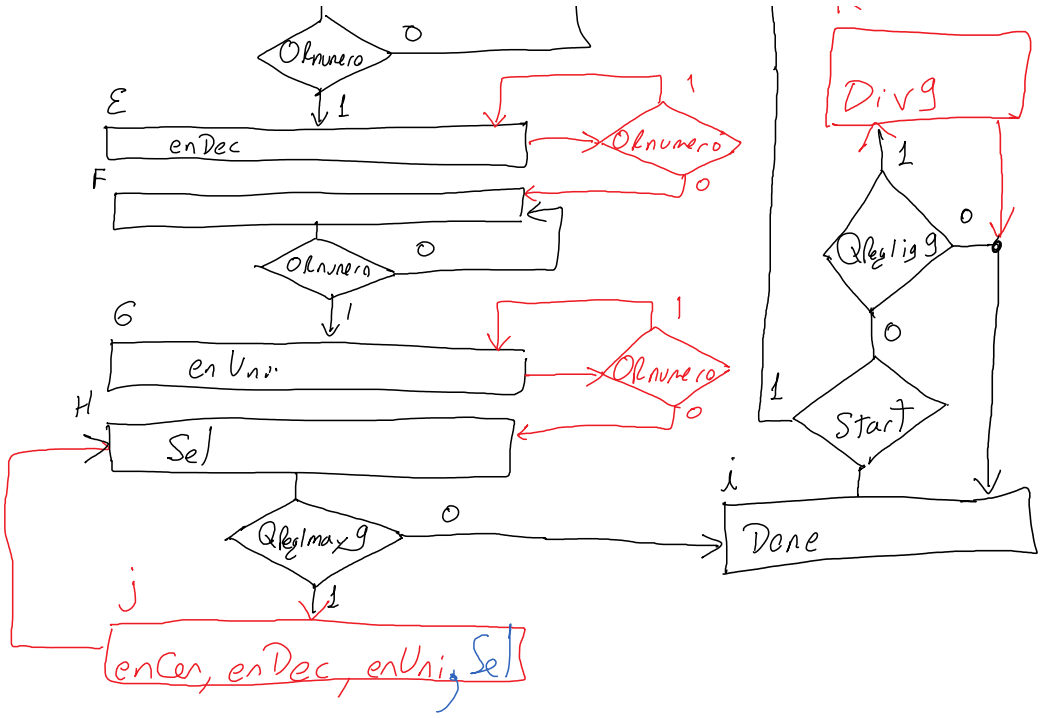





CLK
Numero
10
10
Deco
10
a
BCD
(Prioridad)
ORnumero
C
D
U
Sel
+Vcc
resetn
en
Reg.
Sost.
enDec
Qdec
"0"
enUni
Quni
"8"
clk
Qcnt
Qdec
Quni
Add
4
Qreg1
6
2
8
Deco
Bin
a
BCD

ASM (Loop):


resetn
A
Start
0
1
B
ORnumero
C
enCan
D
K



Loop

X

✓

"0" + "0" + "g" → Bin → g → "0" "0" "g"

"g" + "y" + "g" → Bin → 27 → "2" "7"

X "0" + "2" + "7" → g → "0" "0" "g"

Desafio #2 → VHDL del ASM (cambios)