

Parcial  
con  
Solución

1. Diseñar una FSM Mealy para controlar un motor DC. Diagrama de estados (20%)  
- El motor arranca y gira en sentido normal cuando detecta la secuencia 01101; salida  $Z_1Z_0 = 01$   
- El motor arranca y gira en sentido contrario cuando detecta la secuencia 00010; salida  $Z_1Z_0 = 10$   
- El motor se detiene cuando detecta la secuencia 1010; salida  $Z_1Z_0 = 00$   
- El motor gira en sentido contrario cuando detecta la secuencia 0101 pero primero debe parar antes de cambiar de giro; salida  $Z_1Z_0 = 00^*$ .
2. Diseñar un *data-path* específico y la unidad de control para realizar *división* Data-path y ASM (15%) y prueba de escritorio: dividendo = 11010101 y divisor: 00010101(5%)
3. Diseñar la unidad de control para realizar la operación  $1/X$ . El dato X (8-bit) está en R4, el resultado debe ser almacenado en R5 del procesador UV4. ASM (20%)
4. Diseñar un circuito controlador para realizar el "logaritmo binario" de un dato que se encuentra en R0 y el resultado debe ser almacenado en R1 y R2 del procesador UV4. ASM (20%).
5. Diseñar un circuito controlador para realizar la "multiplicación" usando el algoritmo de Radix 4, los datos están en R4 (multiplicando) y R5 (multiplicador), y el producto debe ser almacenado en R6 (parte alta) y R7 (parte baja) de procesador UV4. ASM (20%)

Ejemplo: multiplicador 1011 y multiplicando 1110:

```

  1 0 1 1
x 1 1 1 0
-----
1 0 0 0 0 1
1 0 0 1 1 0 1 0
-----
1 0 0 1 1 0 1 0

```

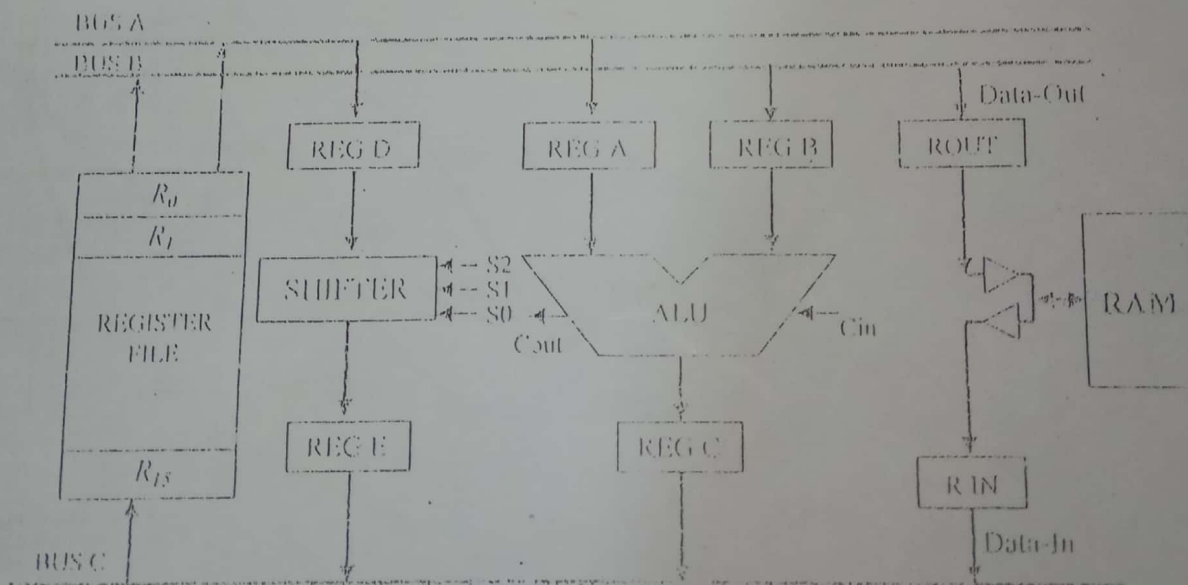


Figura 1. Procesador UV4

Nota:

- ALU ( $S_0S_1$ ): 00 = A or B; 01 = A and B; 10 = A+B; 11 = A-B
- Shifter ( $S_2S_1S_0$ ): 000 = lógico-left (0); 001 = lógico-left (1); 010 = lógico-right (0); 011 = lógico-right (1)  
IXX = transferir.
- Los ASMs solo tendrán tres notas: 5.0 para bueno y óptimo (un número mínimo de estados); 2.5 para bueno pero no óptimo y 1.0 para malo.

213 / 21

10.14 → 1010

Ensayo Tercer Morales - 11/23/32

①

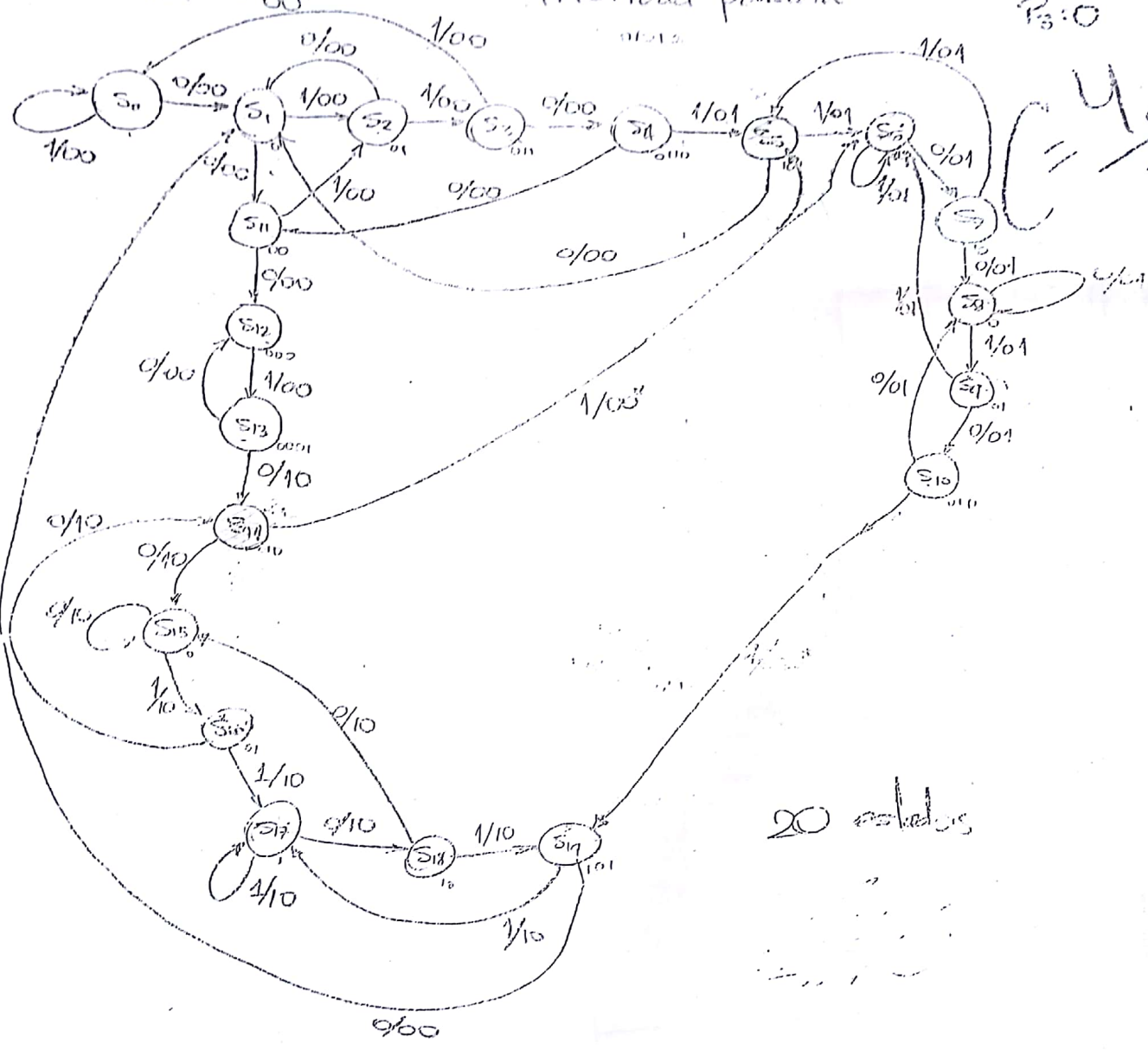
01101 → 01  
 00010 → 10  
 1010 → 00  
 0101 → 00\*

u, s

Al cambiar de tipo  
 Prioridad parada

18

P<sub>3</sub> = 0



20 estados

Nota: Para los ASM específicos cada recálculo contiene lo que se hace en un ciclo de reloj.

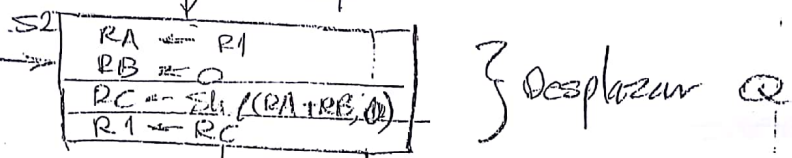
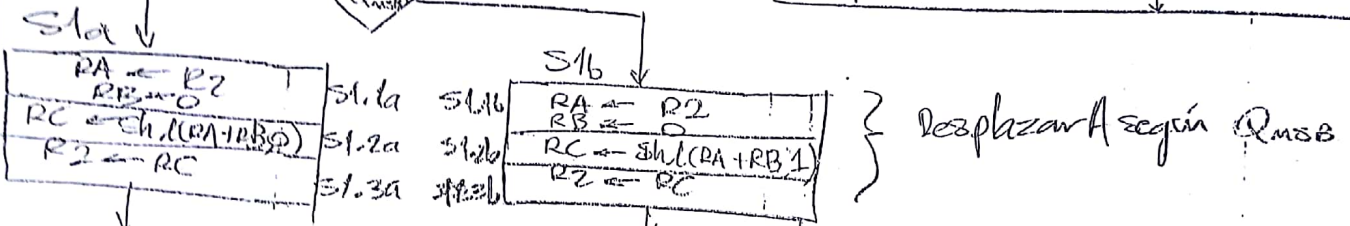
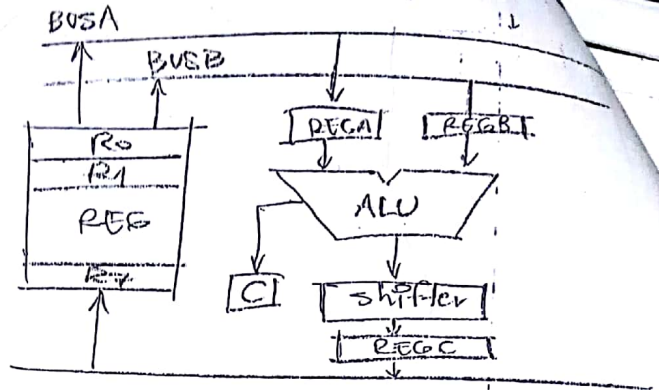
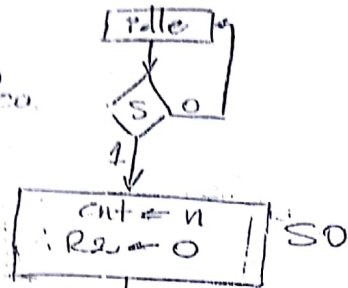
②

ASM:

se supone  $\begin{cases} R_0 \rightarrow \text{dropendo M} \\ R_1 \rightarrow \text{dropendo Q} \\ R_2 \rightarrow \text{Amviador} \end{cases}$  se supone ya guardados

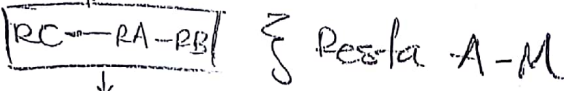
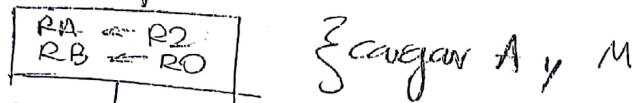
Datapath específico

2.0 No se especifica

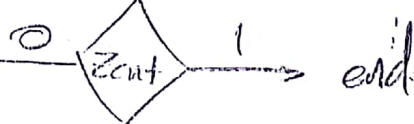
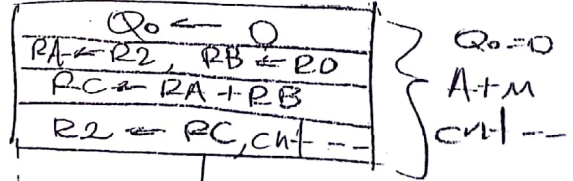
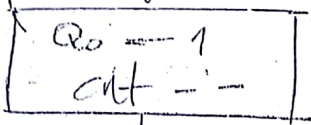


No se especifica por tiempo.

se hace con desplazamientos y teniendo en cuenta USB para que no se pierda



según carry



Determinar Qusb:

$RA \leftarrow R1, RB \leftarrow R1$   
 $RC \leftarrow RA + RB$

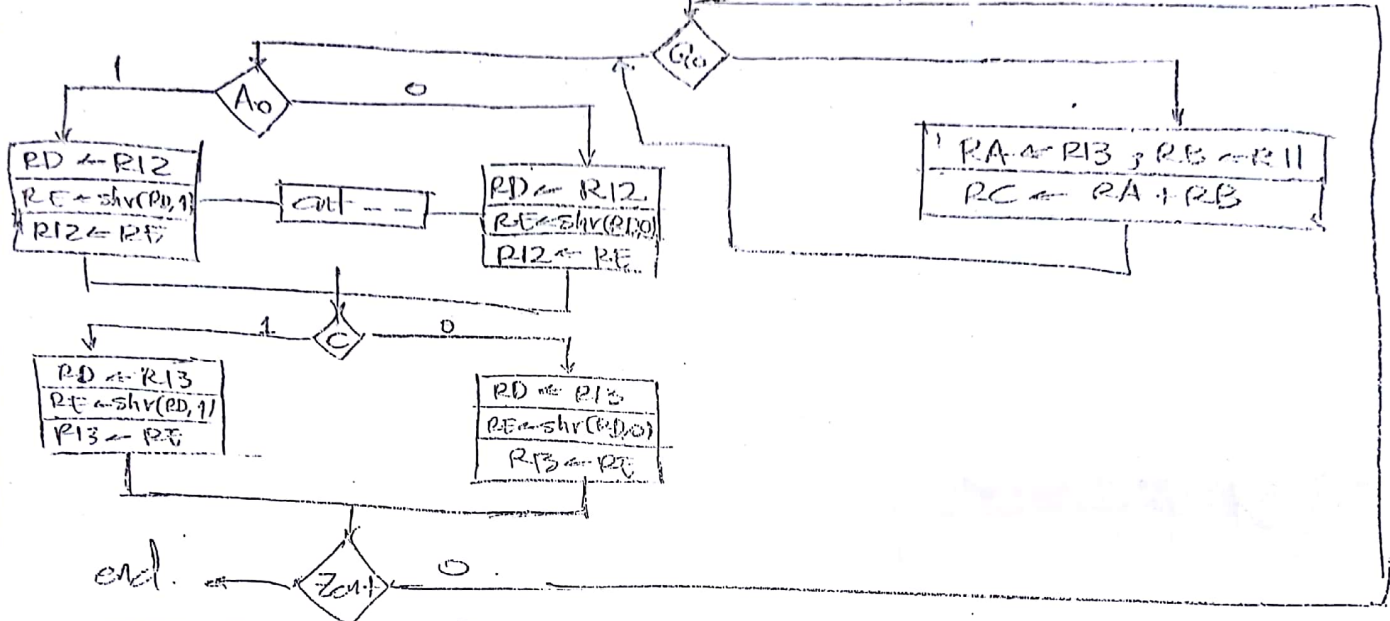
según carry de la ALU:  
if  $c = 1$  then  $Qusb = 1$   
else if  $c = 0$  then  $Qusb = 0$



Producto  $X \times X$ : (para punto 1)

R13  $\leftarrow$  A  
R11  $\leftarrow$  M  
R12  $\leftarrow$  Q

RD $\leftarrow$ R6	RE $\leftarrow$ RD
RM $\leftarrow$ RL	R12 $\leftarrow$ RE
CNT $\leftarrow$ N151, R13 $\leftarrow$ 0	



②  $\frac{1}{2} \times$  Dato en RA

R2  $\leftarrow$  A  
M  $\leftarrow$  1  
R0  $\leftarrow$  Q

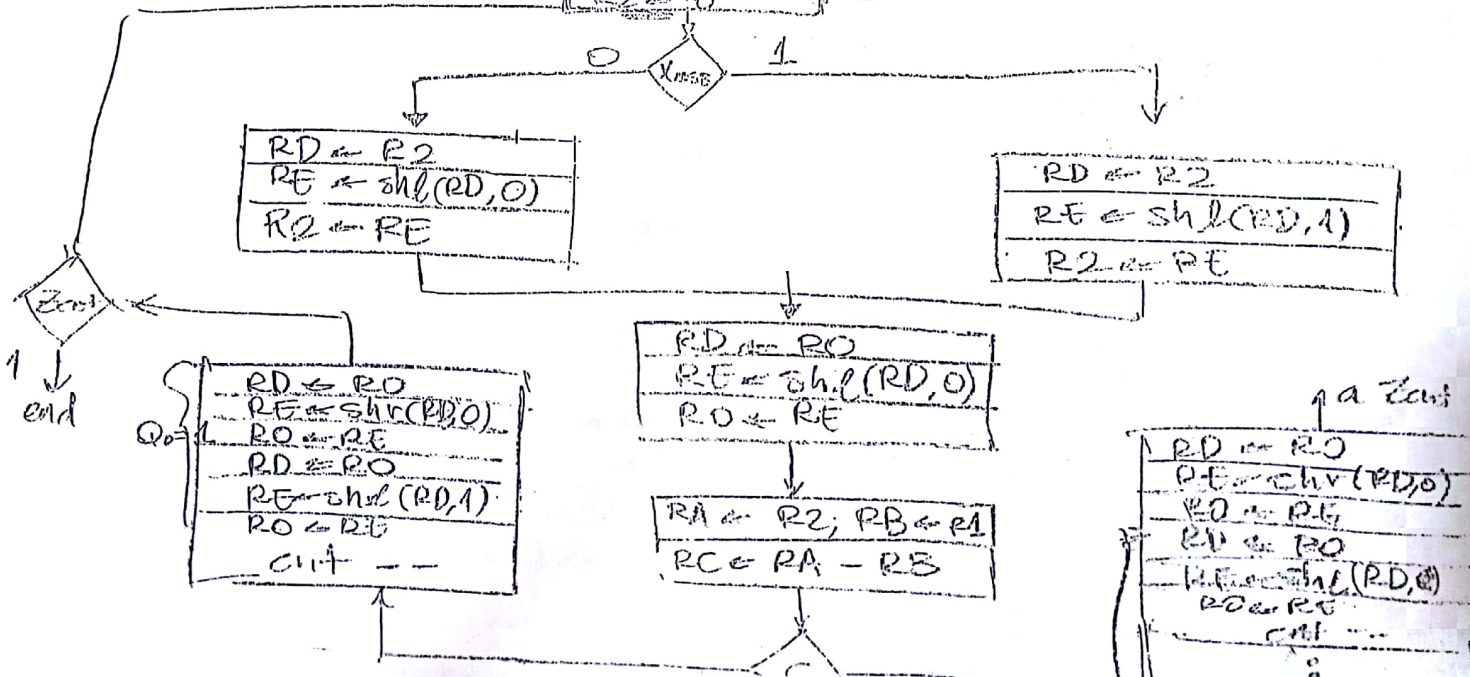
$\frac{1}{2}, 0$

Idle

RD  $\leftarrow$  RA  
RE  $\leftarrow$  RD  
R0  $\leftarrow$  RE

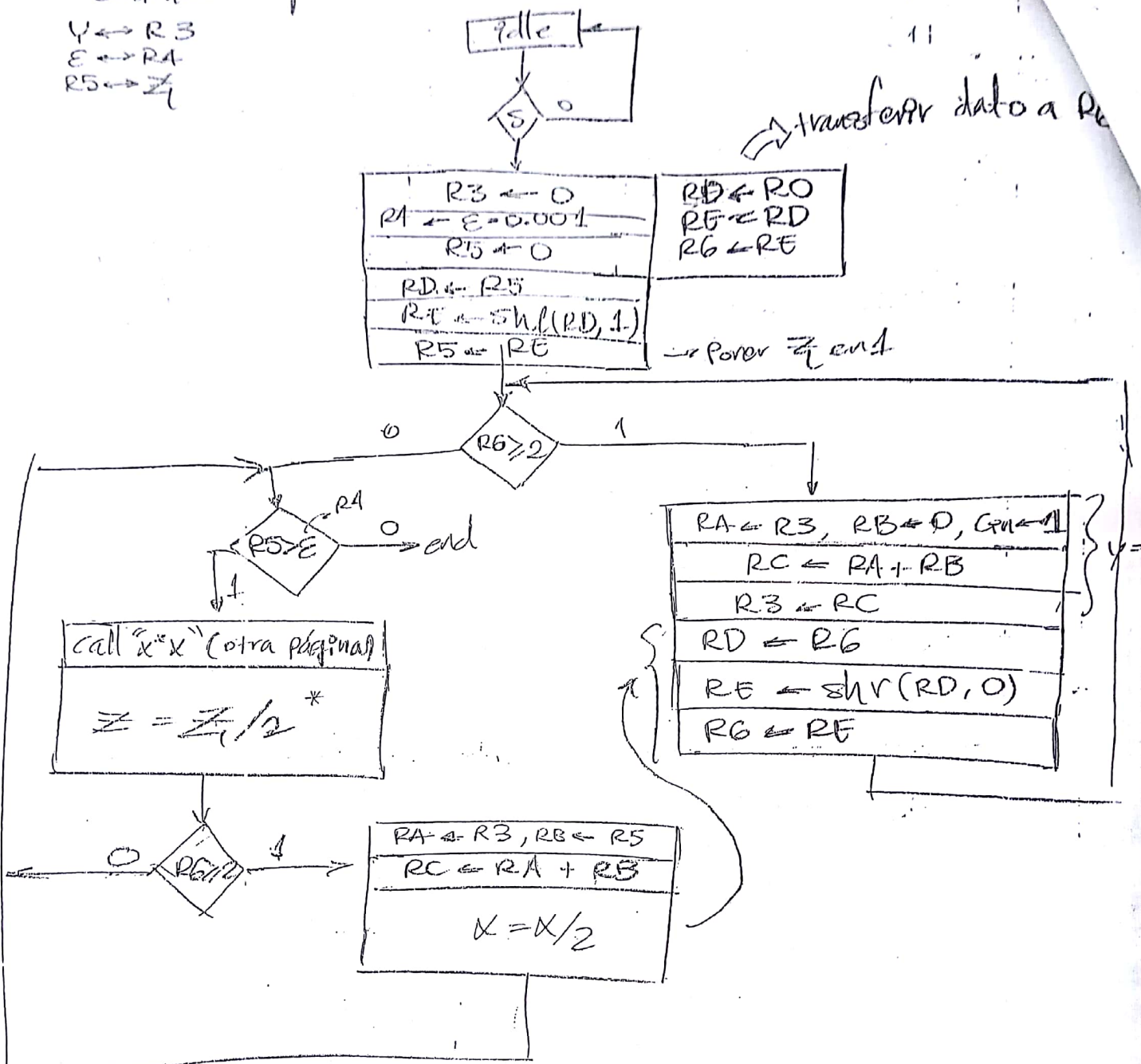
RO $\leftarrow$ RA
CNT $\leftarrow$ N, RD $\leftarrow$ 0
RD $\leftarrow$ shl(RD, 1)
R1 $\leftarrow$ RD
R2 $\leftarrow$ 0

$\frac{1}{2}$  en RA



Continuación punto 4, ASM específico:

Y  $\leftrightarrow$  R3  
E  $\leftrightarrow$  R4  
R5  $\leftrightarrow$  Z



comparacion  $R6 \geq 2$ :

```

R15 ← 0
RD ← R15
RE ← shl(RD, 1)
R15 ← RE
RD ← R15
RE ← shl(RD, 0)
R15 ← RE
RA ← R15, RB ← R6
RC ← R15 - RB
    
```

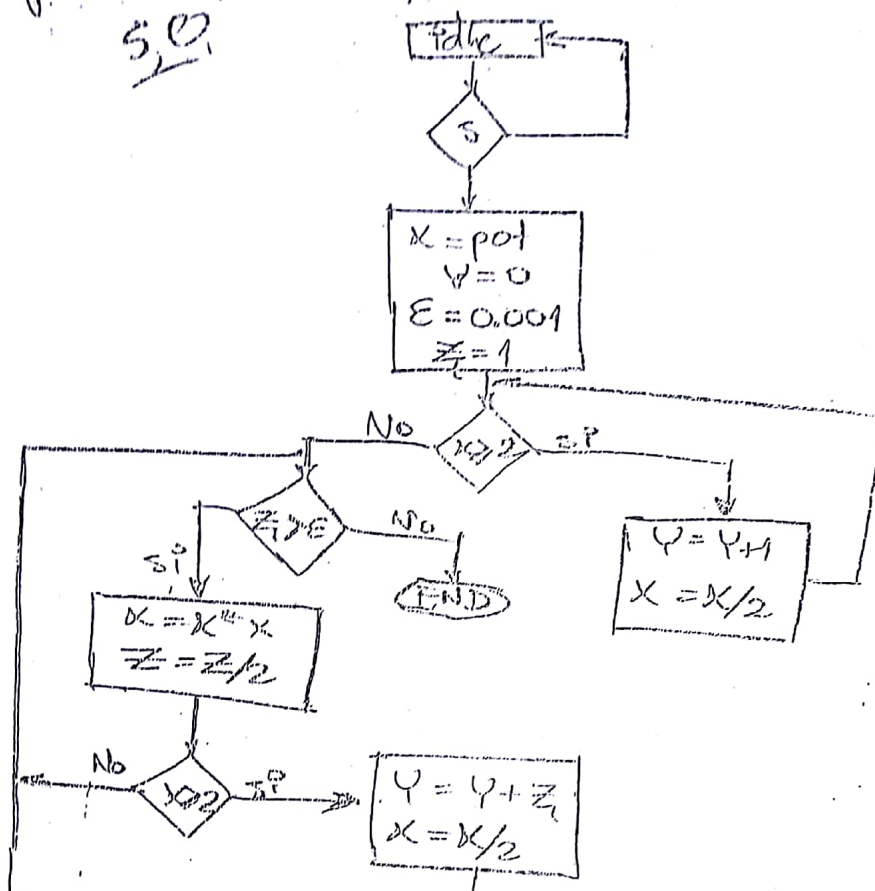
} Poner 2 en R15

// evaluar flag C de la ALU

if C = 1 then  $(R6 \geq 2) = 0$   
else if C = 0 then  $(R6 \geq 2) = 1$

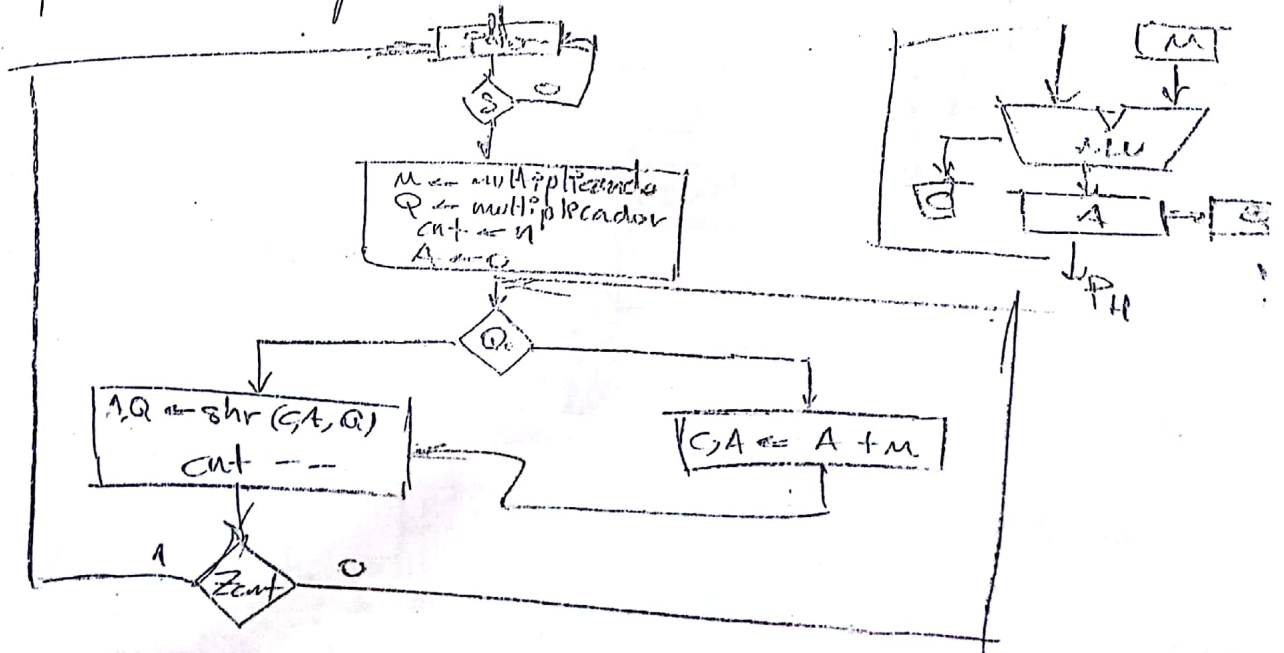
① Algoritmo básico:

5,0



continúa en otra página ASM específico, que sigue este modelo.

Algoritmo para usar en logaritmo:





Sergio Tascón Morales - 1123132

⑤ Paquete 4

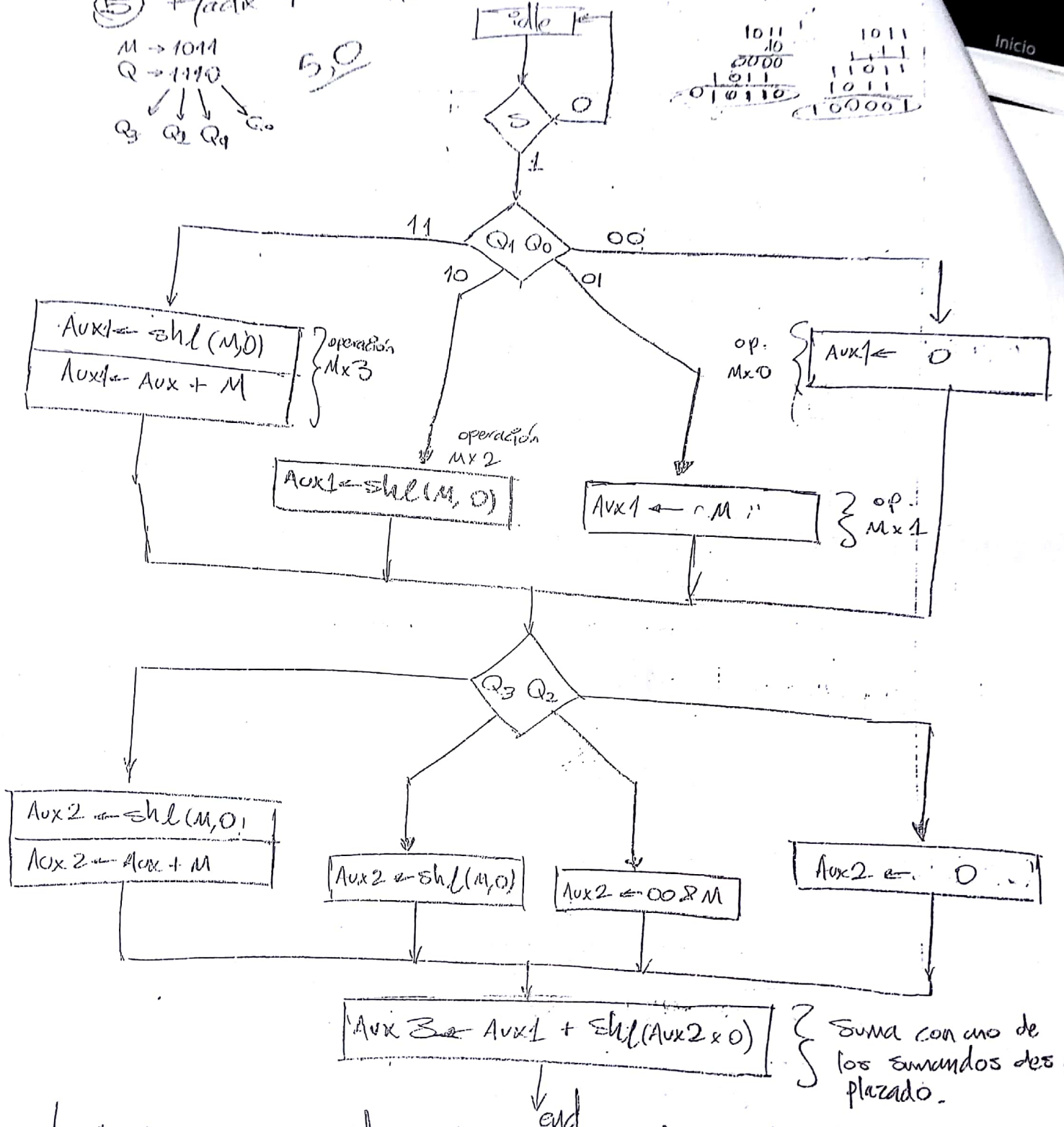
ASM General:

M → 1011  
Q → 1110  
Q<sub>3</sub> Q<sub>2</sub> Q<sub>1</sub> Q<sub>0</sub>

5,0

1011  
10  
0000  
010110  
1011  
11011  
1011  
100001

Inicio

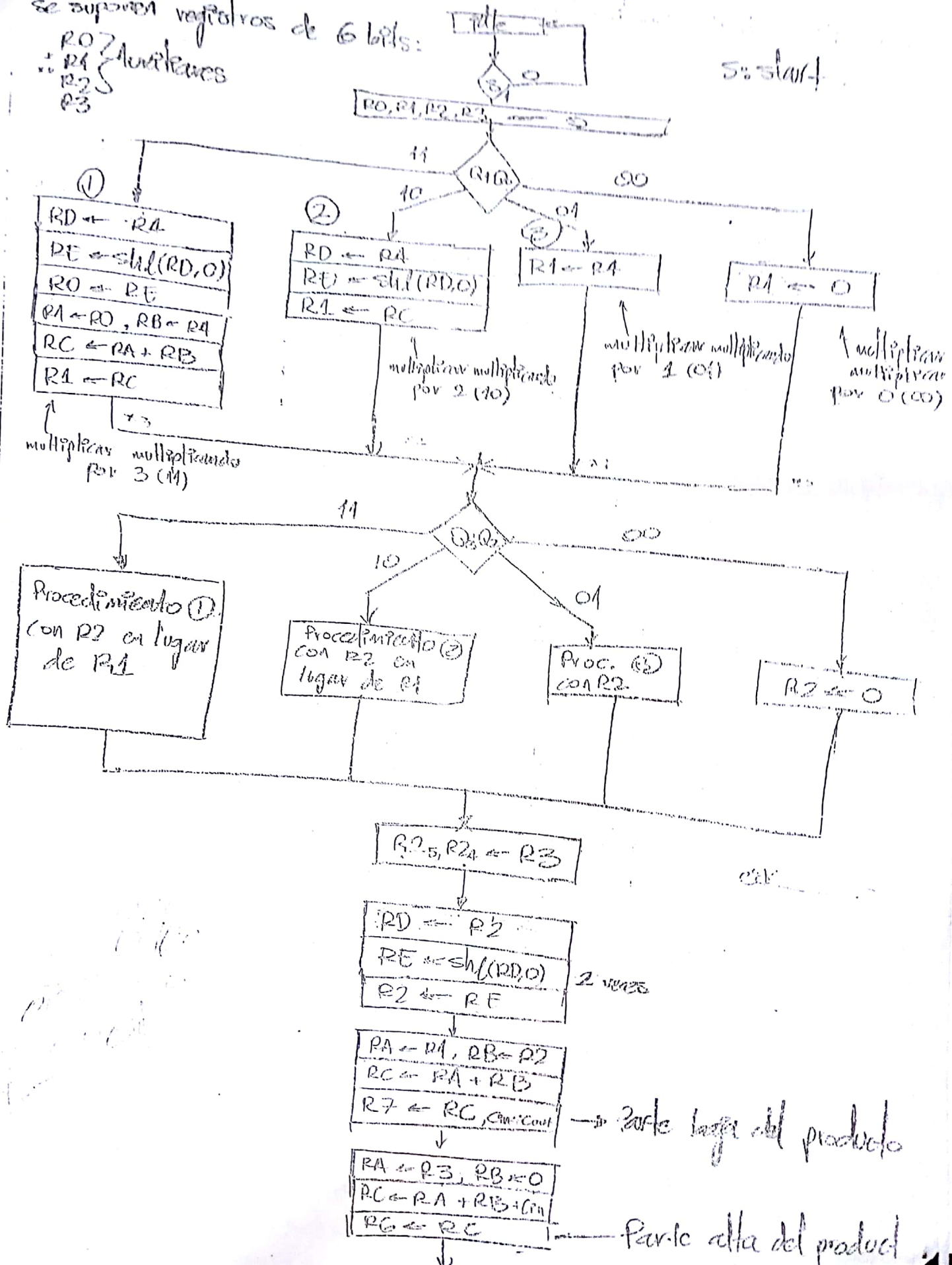


En este ASM no se consideran los tamaños de los registros ni el sentido de bits. En el siguiente ASM se considerará "..."  
Ver "continuación punto 5"

continuación punto D:  
se suponen registros de 6 bits:

RO, R1 } Auxiliares  
R2  
R3

5: start





Sergio Tascón Morales - 1123132

M001010  
000801  
1011

Para el punto 5:

→ Determinación de  $Q_1 Q_0$

→  $Q_0$ :  
 $RD \leftarrow R5$   
 $RE \leftarrow shr(RD, 0)$  // paso 2  
 $RO \leftarrow RE$   
 $RD \leftarrow RO$   
 $RE \leftarrow shl(RD, 1)$   
 $RO \leftarrow RE$   
 $RA \leftarrow R5, RB \leftarrow RO$   
 $RC \leftarrow RA - RB$

según flag Z de la ALU:

if  $Z = 0$  then  $Q_0 = 0$

else if  $Z = 1$  then  $Q_0 = 1$

→  $Q_1$ :

$RD \leftarrow R5$   
 $RE \leftarrow shr(RD, 0)$   
 $R1 \leftarrow RE$   
 $RD \leftarrow R1$

↓ desde aquí se sigue procedimiento seguido para  $Q_0$ , desde "paso 2"

→ Determinación de  $Q_3 Q_2$

→  $Q_3$ :  
 $RD \leftarrow R5$   
 $RE \leftarrow shr(RD, 0)$   
 $RO \leftarrow RE$

secuencia  $\phi$  {  
 $RD \leftarrow RO$   
 $RE \leftarrow shl(RD, 0)$   
 $RO \leftarrow RE$   
 $RA \leftarrow RO, RB \leftarrow RO$   
 $RC \leftarrow RA + RB$

según C de la ALU:

if  $C = 1$  then  $Q_3 = 1$

else if  $C = 0$  then  $Q_3 = 0$

→  $Q_2$ :

$RD \leftarrow R5$   
 $RE \leftarrow shr(RD, 0)$   
 $RO \leftarrow RE$   
secuencia  $\phi$   
secuencia  $\phi$   
 $RA \leftarrow RO, RB \leftarrow RO$   
 $RC \leftarrow RA + RB$

según C de la ALU:

if  $C = 1$  then  $Q_2 = 1$

else if  $C = 0$  then  $Q_2 = 0$