

1. Diseñar **data-path específico de raíz cuadrada** y la unidad de control para calcular $X^{1/2}$. Diagrama lógico de la FSM, RTL-ASM (20%) y prueba de escritorio para este dato = 11101110 (5%)
2. Diseñar la unidad de control para realizar la operación X^6 , el dato X que encuentra en R1, el resultado debe ser almacenada en los registros R2,R3,R4,R5,R6,R7 del **data-path específico del multiplicador**. RTL-ASM (25%)
3. Diseñar un circuito controlador para realizar la multiplicación basada en el algoritmo de **Booth radix-4**. El multiplicando se encuentra en R2, y el multiplicador en R3 del banco de registros del procesador UV4 de la Figura 1. El producto debe ser almacenado en R4 y R5. ASM (20%) y prueba de escritorio, primera iteración: multiplicando = 110101(-11₁₀) y multiplicador = 110101(-11₁₀) (5%).

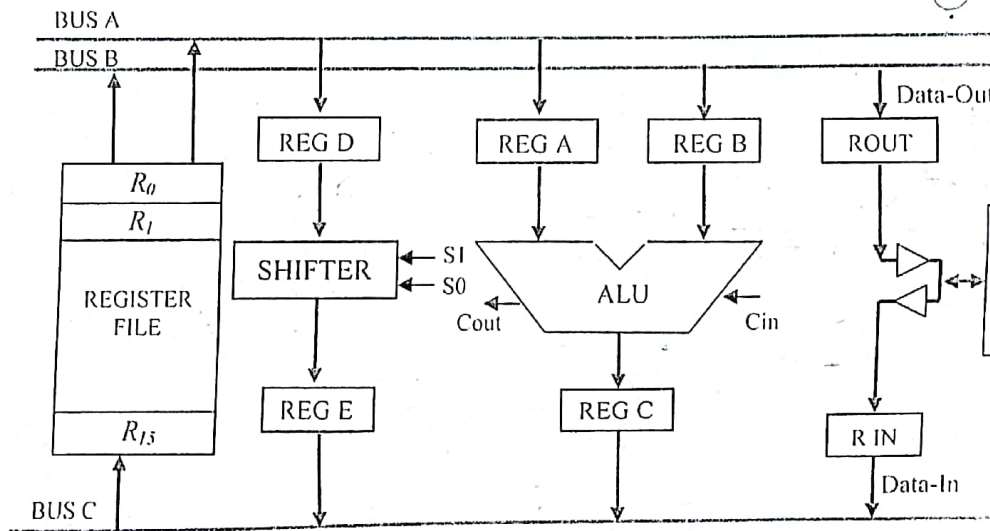


Figura 1: Procesador UV-ARM7

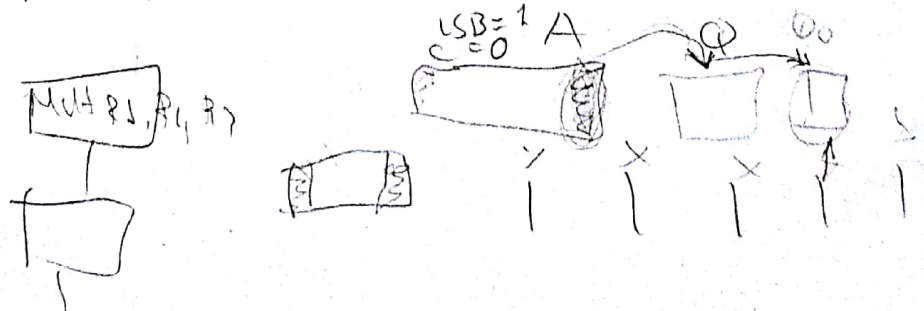
4. Diseñar un circuito controlador para realizar la "multiplicación-rápida" usando el algoritmo de **Radix 8** (multiplica 3-bits simultáneamente), los datos están en R0 (multiplicando) y R1 (multiplicador), y el producto debe ser almacenado en R6 (parte alta) y R7 (parte baja) del procesador UV4 de la Fig. 1. RTL-ASM (20%) y prueba de escritorio, primera iteración: multiplicador = 100111 y multiplicando = 100111(5%)

1	0	1	1
1	1	1	0

0	1	0	1	1	0		
1	0	0	0	0	1		
<hr/>							
1	0	0	1	1	0	1	0

Nota:

- ALU (S₁S₀): 00 = A or B ; 01 = A and B ; 10 = A+B ; 11 = A-B
- Shifter (S₂S₁S₀): 000 = lógico-left {0} ; 001 = lógico-left {1} ; 010 = lógico-right {0} ; 011 = lógico-right {1}
100 = transferir; 101 = aritmético-right {MSB}; 110 = aritmético-left {0}
- 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 diagrama malo.



Paquetes

Paquete I (Sensor LSE)

$$RA \leftarrow R_x ; RB \leftarrow R_y$$

$$RC \leftarrow RA \text{ AND } RB \quad \text{ALU(01)} \quad \checkmark$$

$$R_{aux1} \leftarrow RC \quad \checkmark$$

$$RA \leftarrow R_{aux1} ; RB \leftarrow R_y$$

$$RC \leftarrow RA - RB \quad \checkmark \quad \text{ALU(11)}, C \leftarrow \text{Cont} \quad \checkmark$$

$$S1 \text{ CTD} \leftarrow \text{USE} + 1 \quad \checkmark$$

$$C \leftarrow 1 \quad \checkmark$$

Paquete II (Sumar dos datos)

$$RA \leftarrow R_x ; RB \leftarrow R_y$$

$$RC \leftarrow RA + RB \quad \text{ALU(10)} \quad \checkmark$$

$$(R_x) \leftarrow RC ; C \leftarrow \text{Cont} \quad \checkmark$$

Paquete IV (desplazar derecha aritmetico)

$$RD \leftarrow R_x$$

$$RE \leftarrow \text{Ashr}(RD) \quad \text{sh(101)} \quad \checkmark$$

$$(R_x) \leftarrow RE \quad \checkmark$$

Paquete VI (despla derecha 2)

$$RD \leftarrow R_x$$

$$RE \leftarrow \text{shr}(RD, 2) \quad \text{sh(011)} \quad \checkmark$$

$$R_x \leftarrow RE$$

Paquete VIII (despl 299 0)

$$RD \leftarrow R_x$$

$$RE \leftarrow \text{shl}(RD, 0) \quad \text{sh(000)} \quad \checkmark$$

$$R_x \leftarrow RE$$

Paquete IX (despl 299 1)

$$RD \leftarrow R_x$$

$$RE \leftarrow \text{shl}(RD, 1) \quad \text{sh(001)} \quad \checkmark$$

$$R_x \leftarrow RE$$

Paquete III (restar dos datos)

$$RA \leftarrow R_x ; RB \leftarrow R_y$$

$$RC \leftarrow RA - RB \quad \text{ALU(10)} \quad \checkmark$$

$$(R_x) \leftarrow RC ; C \leftarrow \text{Cont} \quad \checkmark$$

Paquete V (desplazar derecha 1mm)

$$RD \leftarrow R_x$$

$$RC \leftarrow \text{shr}(RD) \quad \text{sh(010)} \quad \checkmark$$

$$(R_x) \leftarrow RC \quad \checkmark$$

Paquete VII (transferir dato - hacer copia)

$$RA \leftarrow R_x ; RB \leftarrow 0$$

$$RC \leftarrow RA + RB \quad \text{ALU(10)} \quad \checkmark$$

$$(R_y) \leftarrow RC \quad \checkmark$$

Paquete IX (Sensor JASE)

$$RA \leftarrow R_x ; RB \leftarrow R_x \quad \checkmark$$

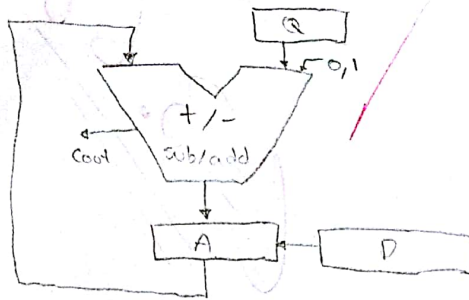
$$RC \leftarrow RA + RB \quad \text{ALU(10)} \quad \checkmark \quad C \leftarrow \text{Cont}$$

$$C \leftarrow 1 \quad \checkmark$$

OK

15.0 5.0 (b)
21.5 5.0 (b)
35.0 5.0 (b)
44.0 5.0 (b)

17. Data-path específico raíz cuadrada.

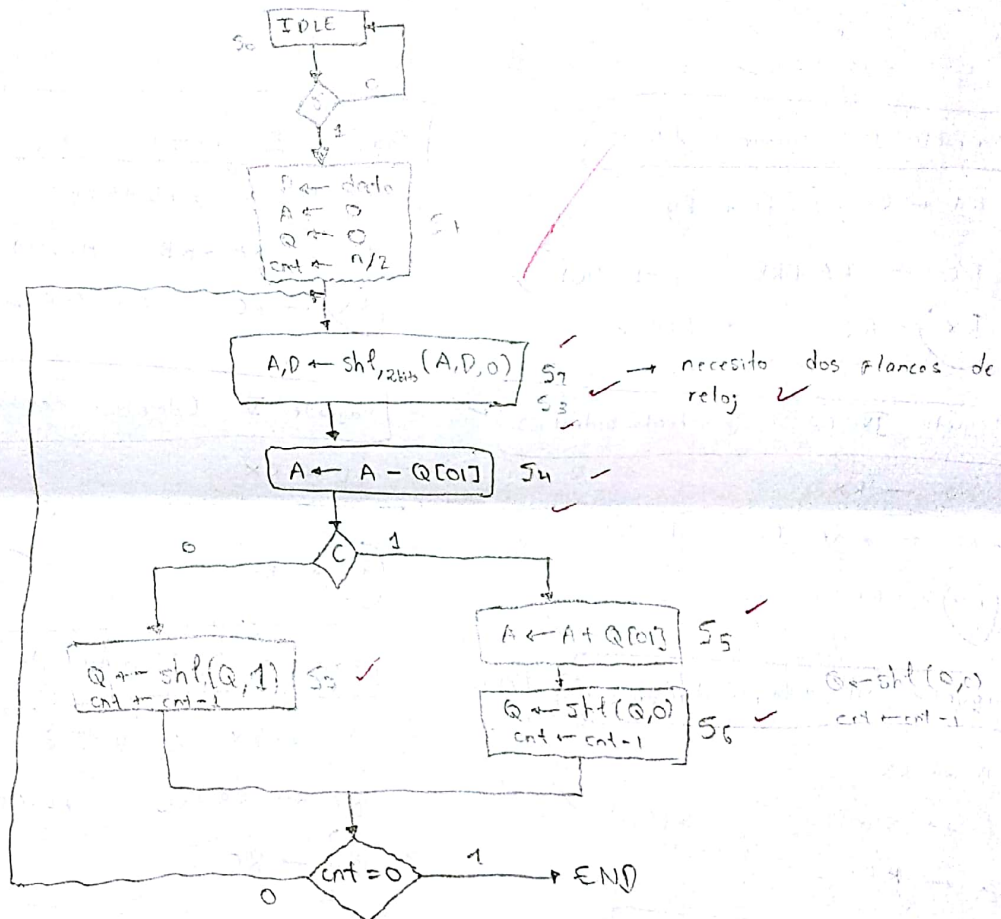


D = radicando

A = acumulador

Q = raíz

Los registros desplazan un solo bit en cada flanco de reloj.



RTL.

S1: $D \leftarrow \text{dato}$; $A \leftarrow 0$ $Q \leftarrow 0$ $\text{cnt} \leftarrow n/2$; clrA clrQ loadD

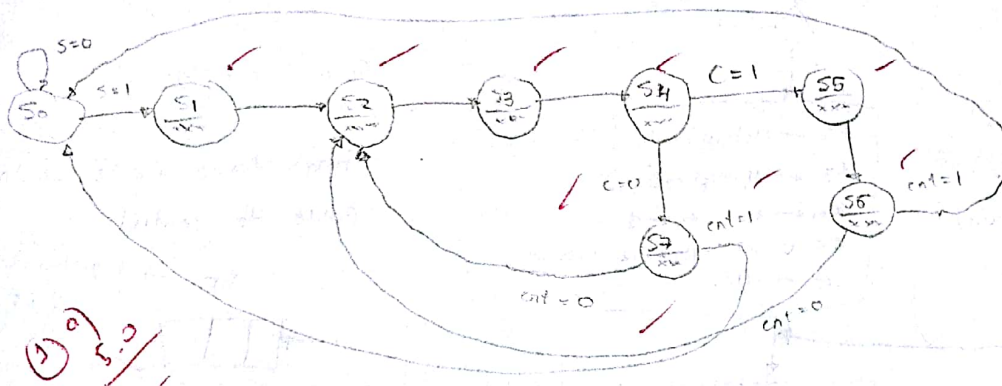
S2: $A, D \leftarrow \text{shl}(A, D, 0)$ ✓ ; shA , shD
S3: " " " "

S4: $A \leftarrow A - Q[\text{cnt}]$; $\text{c} \leftarrow \text{Cont.}$; wRA

S5: $A \leftarrow A + Q[\text{cnt}]$; wRA .

S6: $Q \leftarrow \text{shl}(Q, 0)$; shQ_0

S7: $Q \leftarrow \text{shl}(Q, 1)$; shQ_1



In			EP			PE			Salidas								
S	C	cnt							clrA	clrQ	loadD	sha	shD	c	wra	sh0	sh1
0	x	x	0	0	0	0	0	0									
1	x	x	0	0	0	0	0	1									
x	x	x	0	0	1	0	1	0	1	1	1						
x	x	x	0	1	0	0	1	1				1	1				
x	x	x	0	1	1	1	0	0				1	1				
x	0	x	1	0	0	1	1	1						1	1		
x	1	x	1	0	0	1	0	1									
x	x	x	1	0	1	1	1	0								1	
x	x	0	1	1	0	0	1	0								1	
x	x	1	1	1	0	0	0	0									1
x	x	0	1	1	1	0	1	0									1
x	x	1	1	1	1	0	0	0									
			Q ₂	Q ₁	Q ₀	P ₂	P ₁	P ₀									

En lugar de hacer mapas de Karnaugh es mejor trabajar con minterminos

$$D_0 = S \bar{Q}_0 \bar{Q}_1 \bar{Q}_2 + \bar{Q}_0 \bar{Q}_1 \bar{Q}_2 + \bar{Q}_0 \bar{Q}_1 Q_2$$

$$D_1 = Q_0 \bar{Q}_1 \bar{Q}_2 + \bar{Q}_0 \bar{Q}_1 \bar{Q}_2 + \bar{Q}_0 \bar{Q}_1 Q_2 \bar{c} + Q_0 \bar{Q}_1 Q_2 + \bar{Q}_0 Q_1 Q_2 \bar{cnt} + Q_0 Q_1 Q_2 \bar{cnt}$$

$$P_2 = Q_0 Q_1 \bar{Q}_2 + \bar{Q}_0 \bar{Q}_1 Q_2 + Q_0 \bar{Q}_1 Q_2$$

Salidas solo dependen estado presente

$$clrA, clrQ, loadD = Q_0 \bar{Q}_1 \bar{Q}_2$$

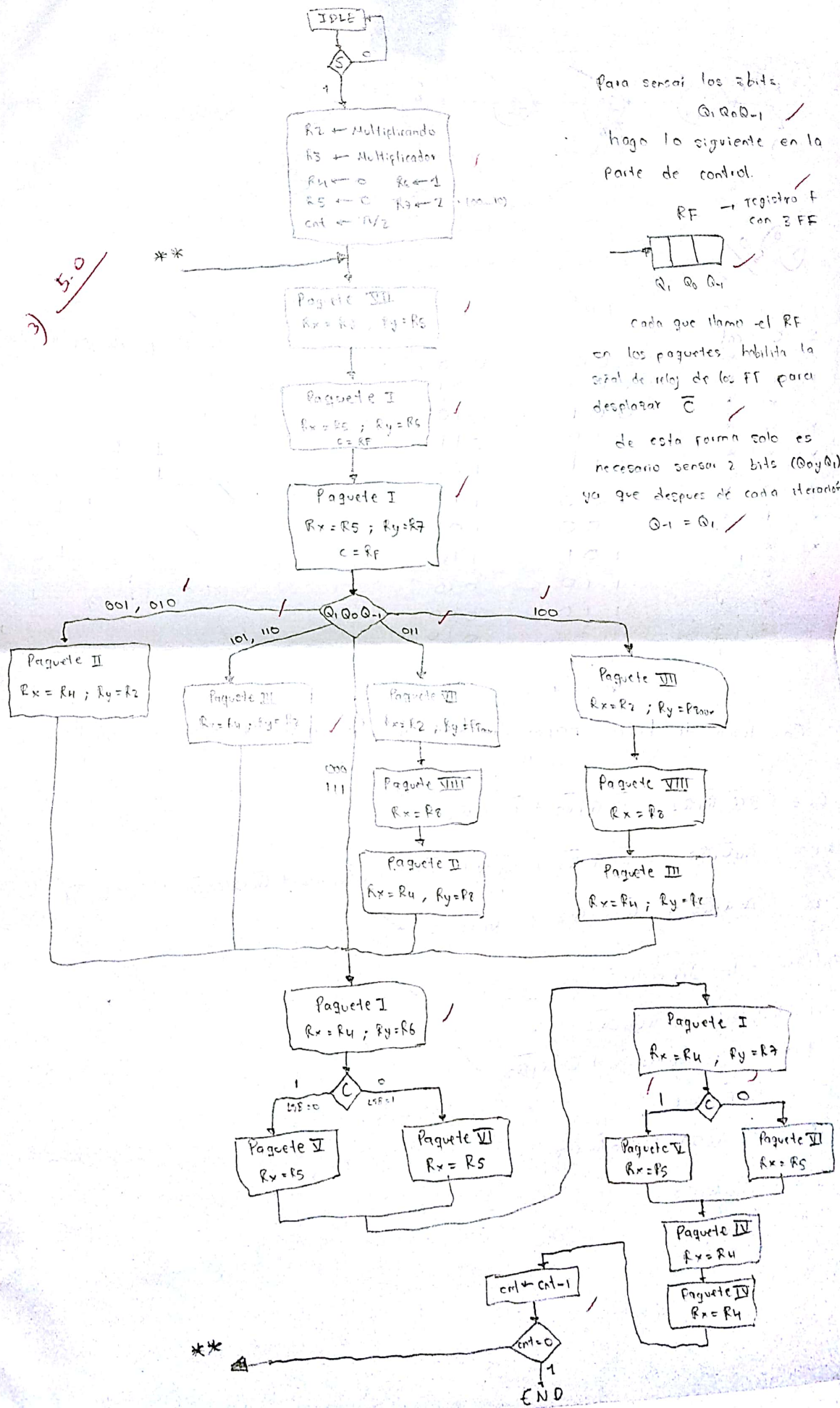
$$sha, shD = \bar{Q}_0 Q_1 \bar{Q}_2 + Q_0 \bar{Q}_1 \bar{Q}_2$$

$$C = \bar{Q}_0 \bar{Q}_1 Q_2$$

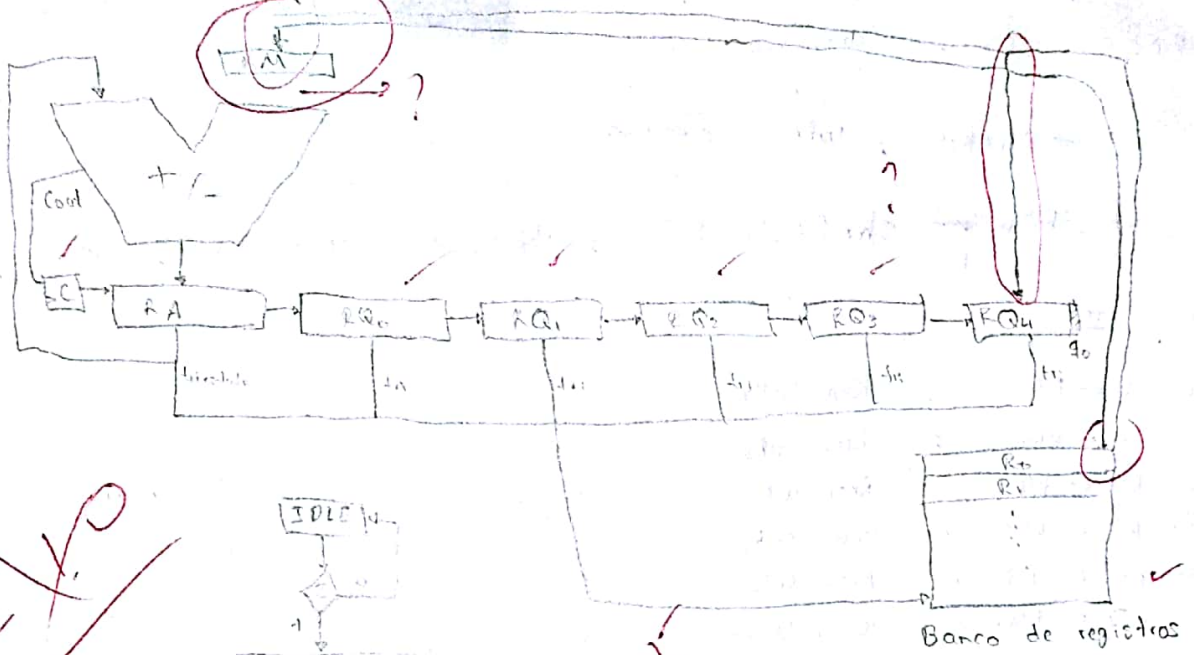
$$wra = \bar{Q}_0 \bar{Q}_1 Q_2 + Q_0 \bar{Q}_1 Q_2$$

$$sh0 = \bar{Q}_0 Q_1 Q_2$$

$$sh1 = Q_0 Q_1 Q_2$$



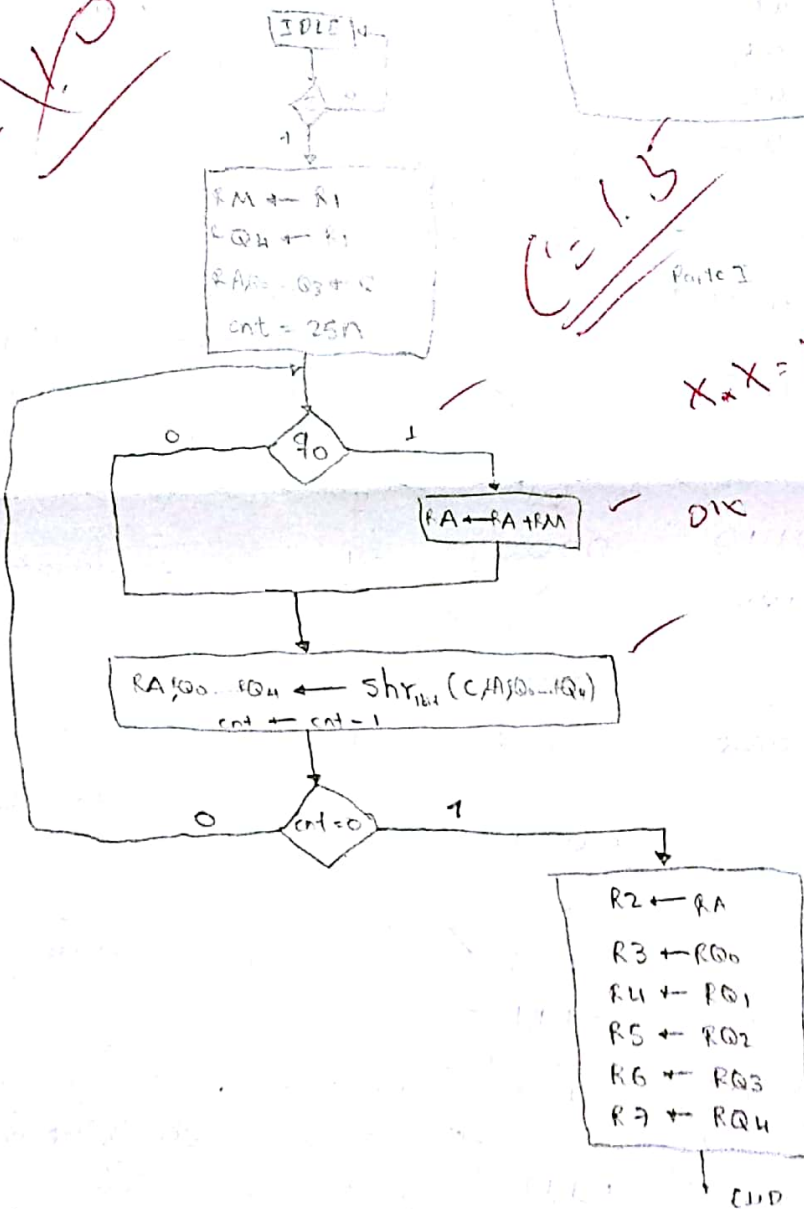
Punto 2. Datapath específico



~~C = 1.5~~

~~C = 1.5~~
Parte I

~~X * X = X^2~~



Parte II

RTL.

F42550 Parte I

normal

F42550 blvd 200

$RM \leftarrow R1$; WRM
 $RQ4 \leftarrow R1$; $WRQ4$
 $RA \dots RQ3 \leftarrow 0$; $clRA, RQ3$

2. $RA \leftarrow RA + RM$; WRA $C \leftarrow Cont.$

3. $RA \dots RQ4 \leftarrow shr(C, RA \dots RQ4)$; $shenble RA \dots shenble RQ4$; $C \leftarrow 0$.

Parte II

1 $R2 \leftarrow RA$; $RRA WR2$
 2 $R3 \leftarrow RQ0$; $RQ0 WR3$
 3 $R4 \leftarrow RQ1$; $RQ1 WR4$
 4 $R5 \leftarrow RQ2$; $RQ2 WR5$
 5 $R6 \leftarrow RQ3$; $RQ3 WR6$
 6 $R7 \leftarrow RQ4$; $RQ4 WR7$

Pruebas de escritorio.

A = acumulador

D = radicando

4 iteraciones

Q = raíz.

raíz. Dato = 11101110.

A	D	Q	q	
00000000	11101110	0000	01	(1) iteración
00000011	10111000			
- 00000001				
00000010		0001		
00001010	11100000	0001	01	
- 00000101				
00000101		0011	01	(2) iteración
00010111	10000000	0011	01	
- 00001101				
00010101		0111	01	(3) iteración
00101010	00000000	0111	01	
- 00011101				
00001101		1111		(4) iteración
residuo.		raíz		

1111
 raíz

Luis David Gutiérrez Loaiza.

cod: 1523647.

Prueba escritorio radix-4 Booth

Usando los datos con 8 bits.

Multiplicando = 110101 (-11₁₀)

Multiplicador = 110101 (-11₁₀)

A = acumulador

D = multiplicador

M = multiplicando

11110101

11110101

2A = 11101010

-A = 00001011

-2A = 00010110

4 iteraciones

A

D

M

00000000
11110101

1111010101
+A

11110101

① iteración

11110101

11111101

0111110110

11111101

11110101

11110010

11111100

1001111110

② iteración

11111100

+ 00001011

00000111

00000001

111001111

③ iteración

00000000

01111001

0 (solo deslazar)

→ (121₁₀)

Parte alta

Parte baja

Prueba de escritorio Radix 8.

multiplicando = 100111

multiplicador = 100111

$$y = (0 \dots 100)$$
