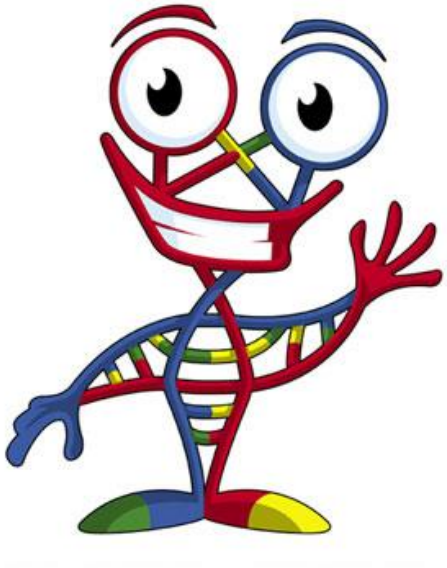
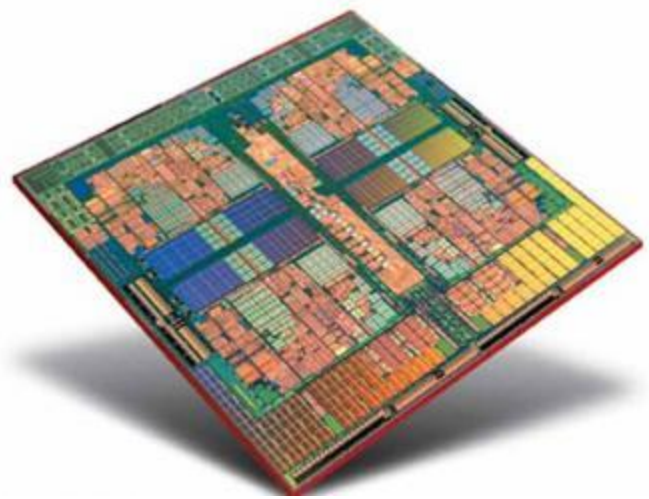
Digital System Design Course

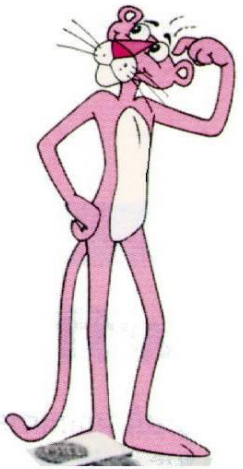
Professor: Jaime Velasco-Medina

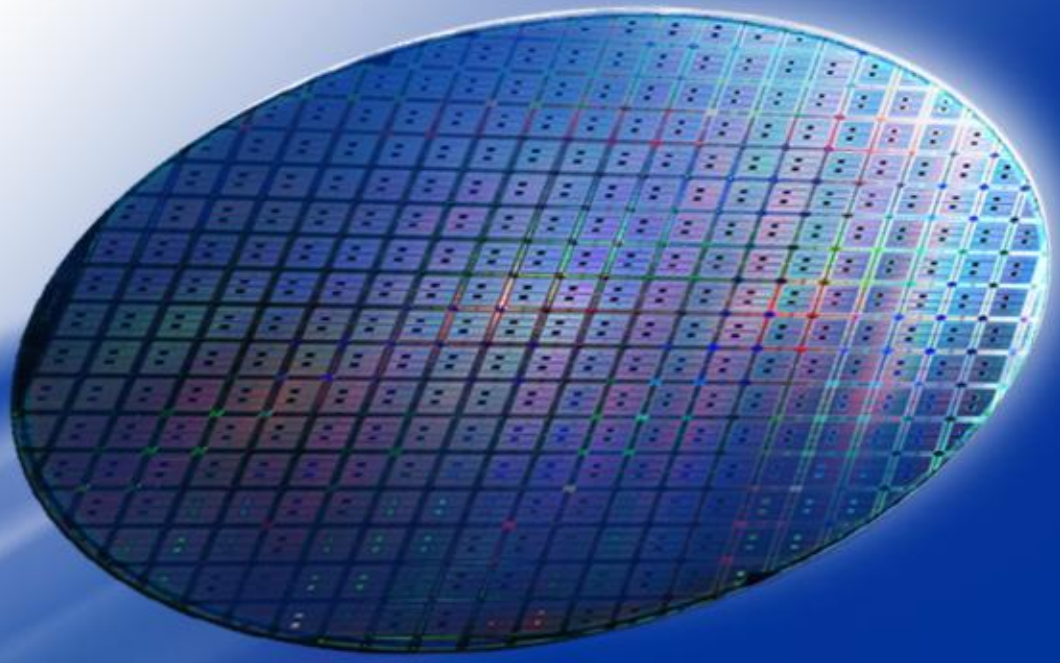
Bionanoelectronics Group

Digital System Design Course



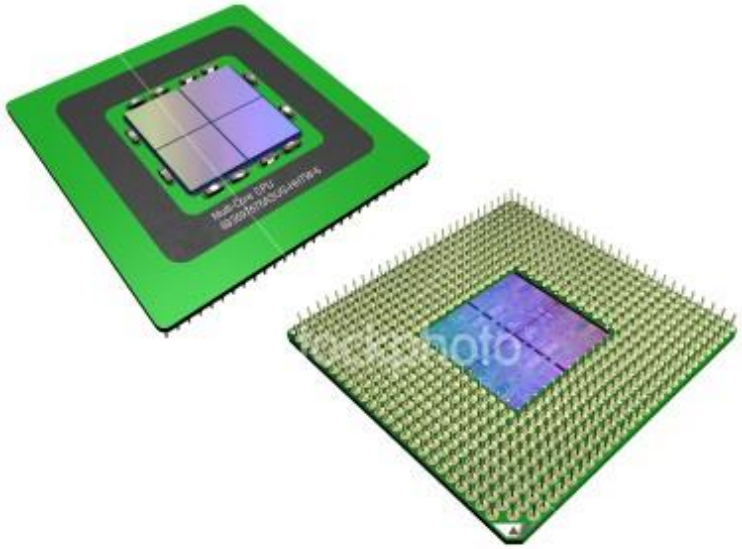






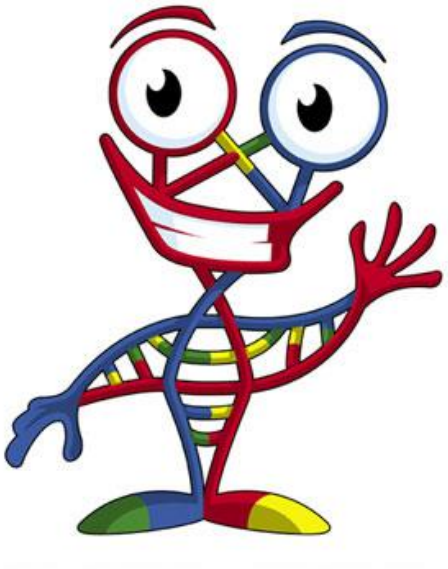
Digital System Design Course

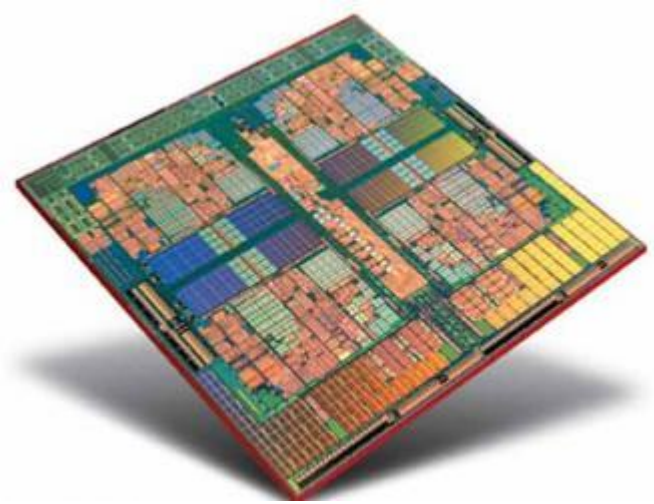
Finite States Machine Design



Lecture 10-0

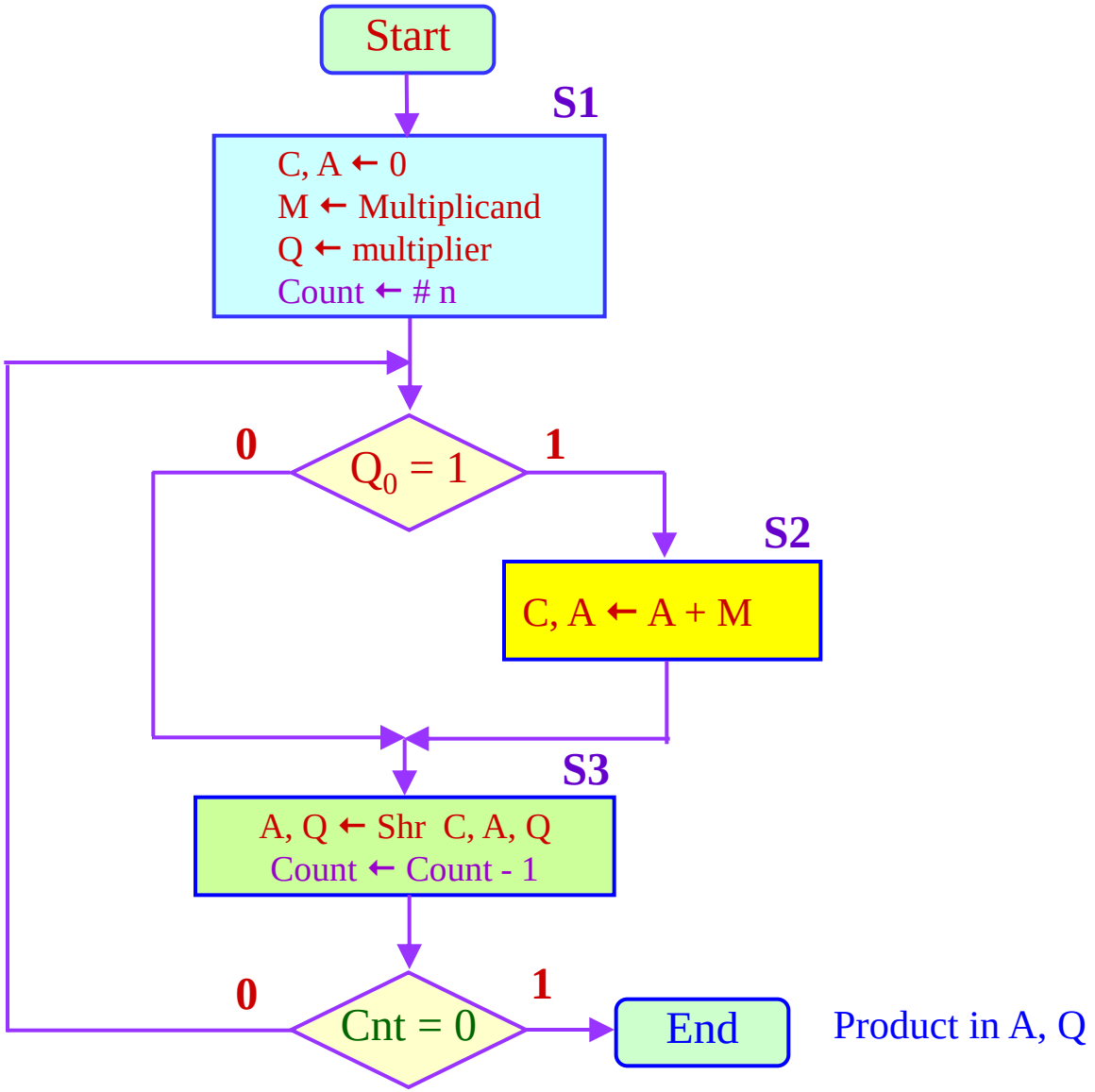
Multiplier Design





1. Binary Multiplier

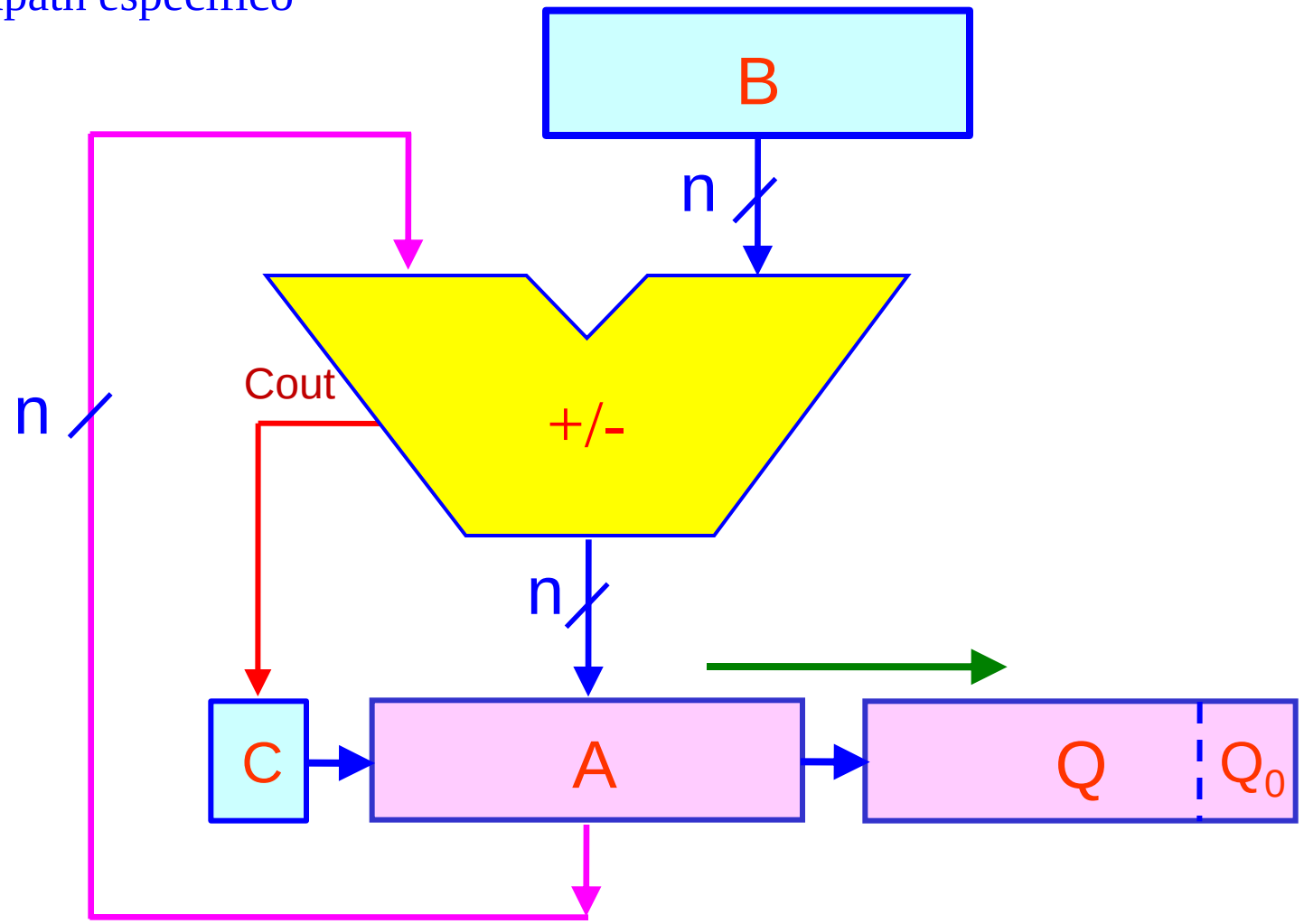
algorithm



1. Binary Multiplier

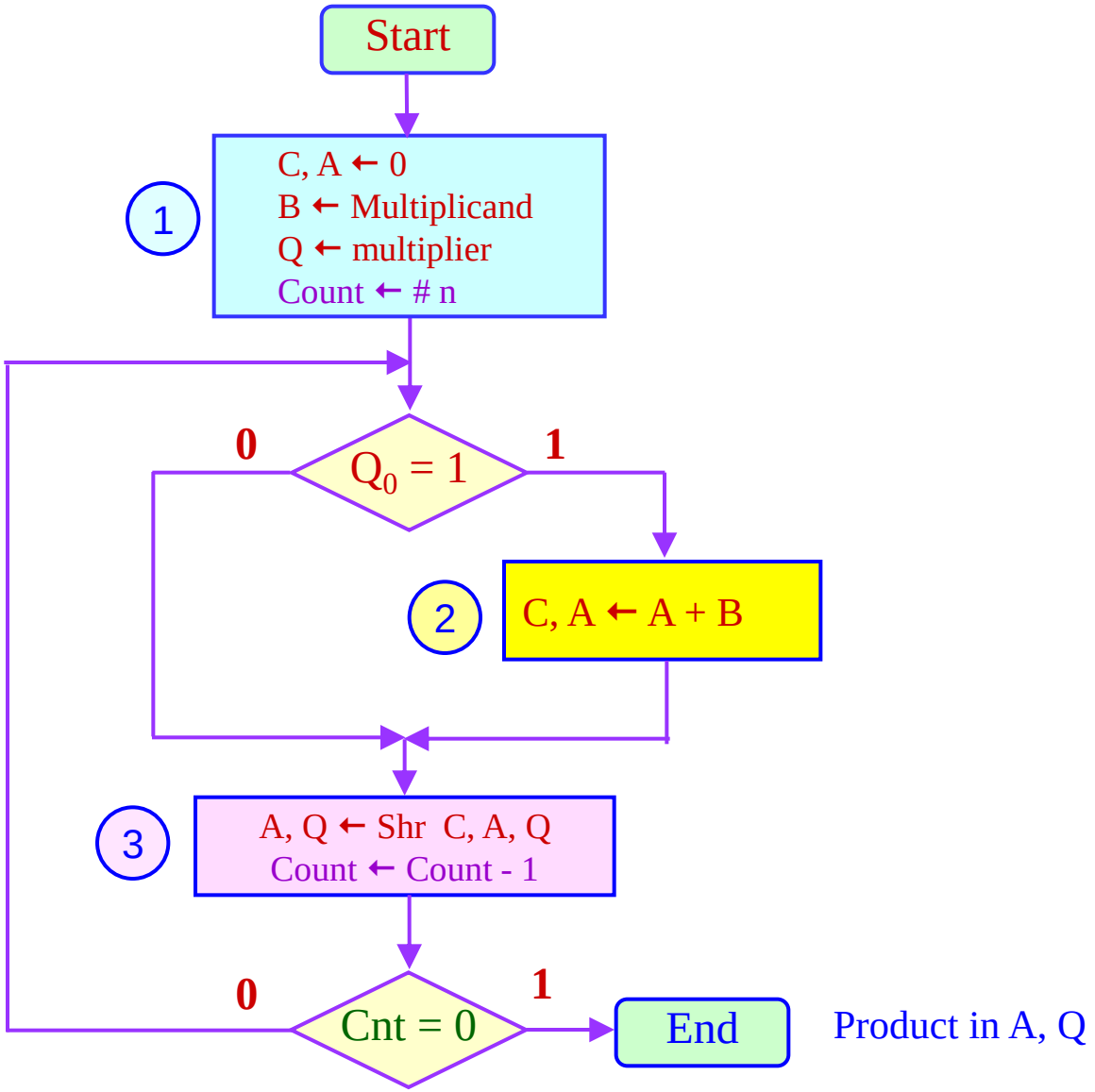
Universidad del Valle

Datapath especifico



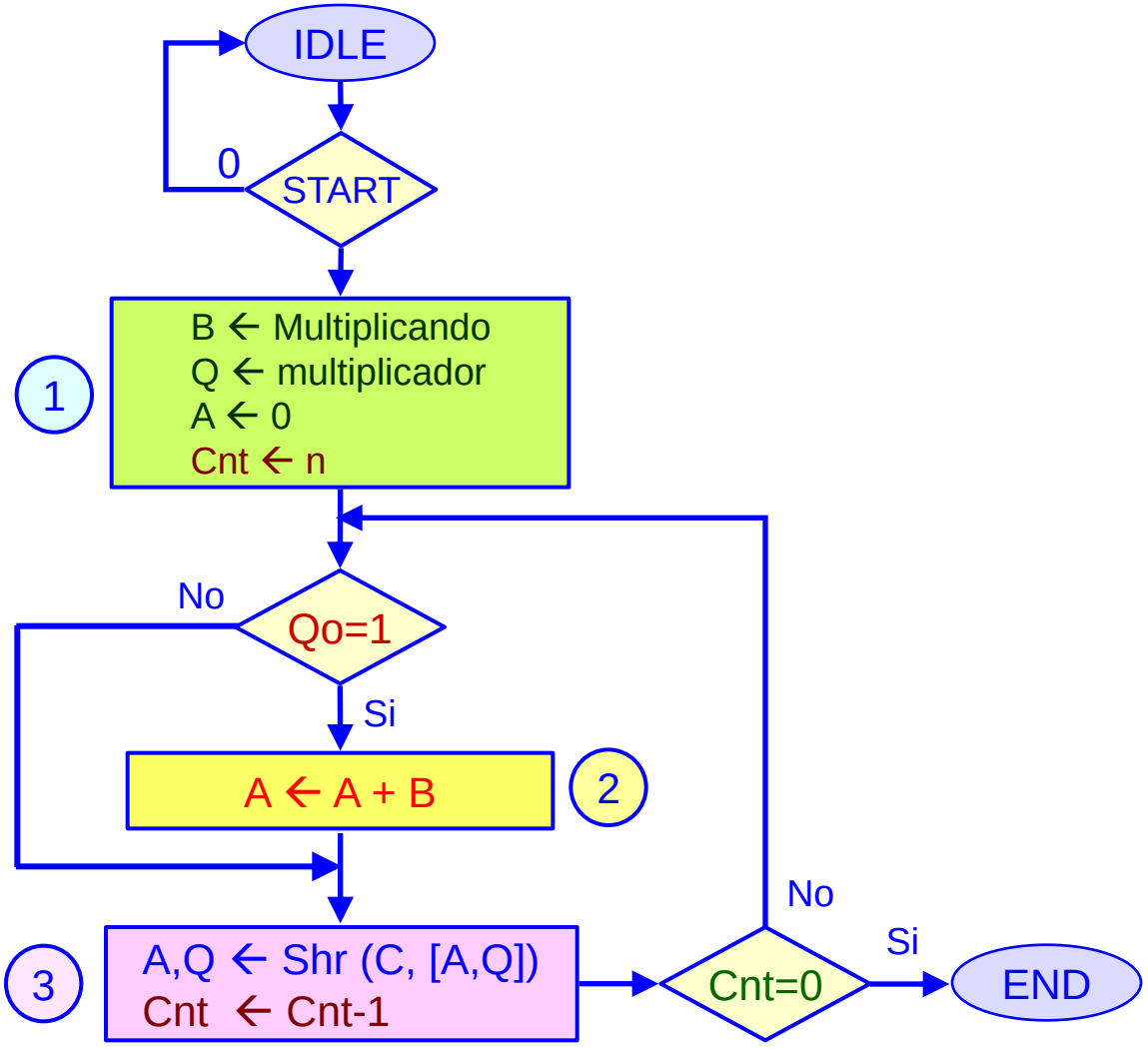
1. Binary Multiplier

ASM diagram



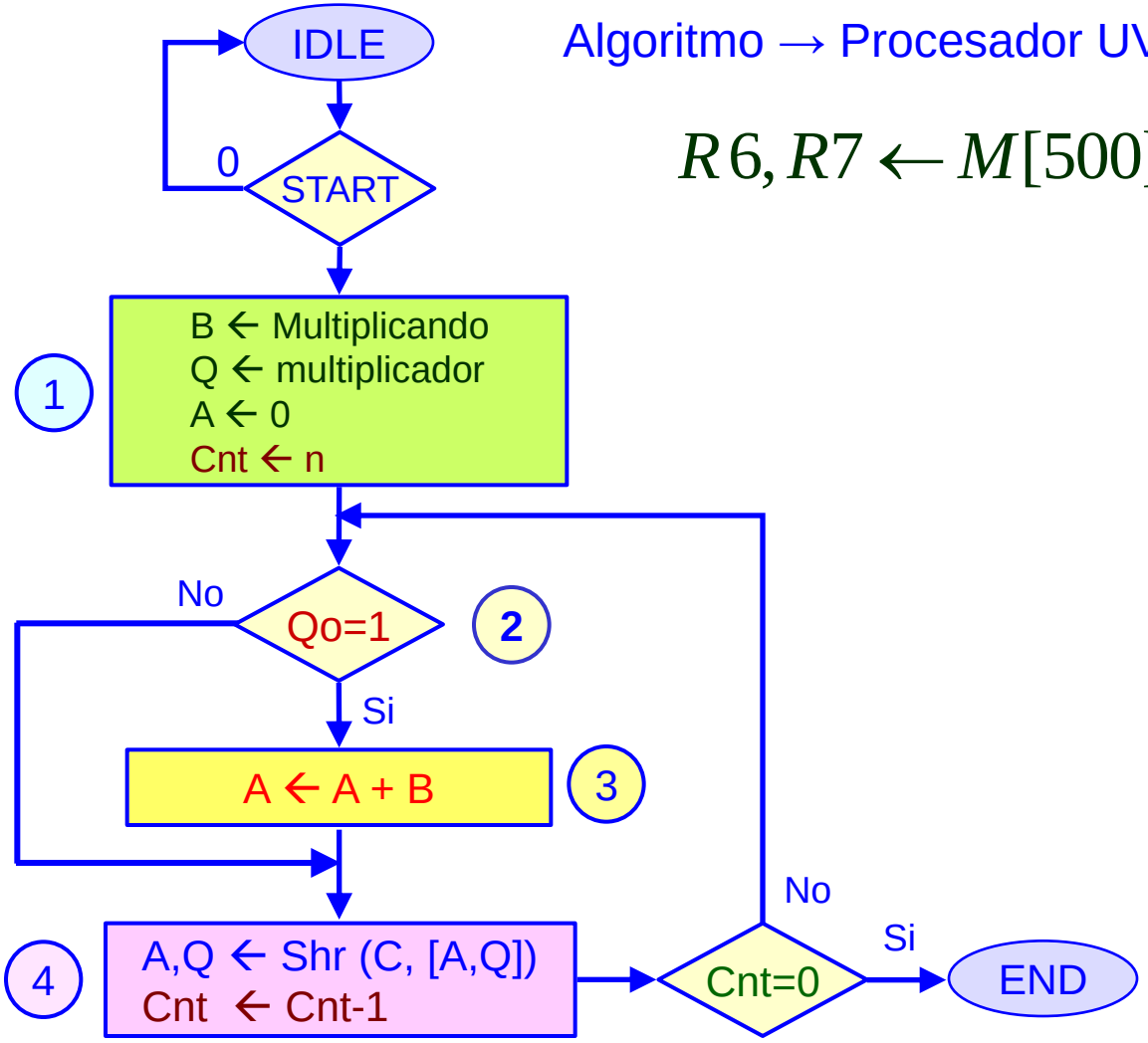
Multiplication

ASM diagram



Multiplication

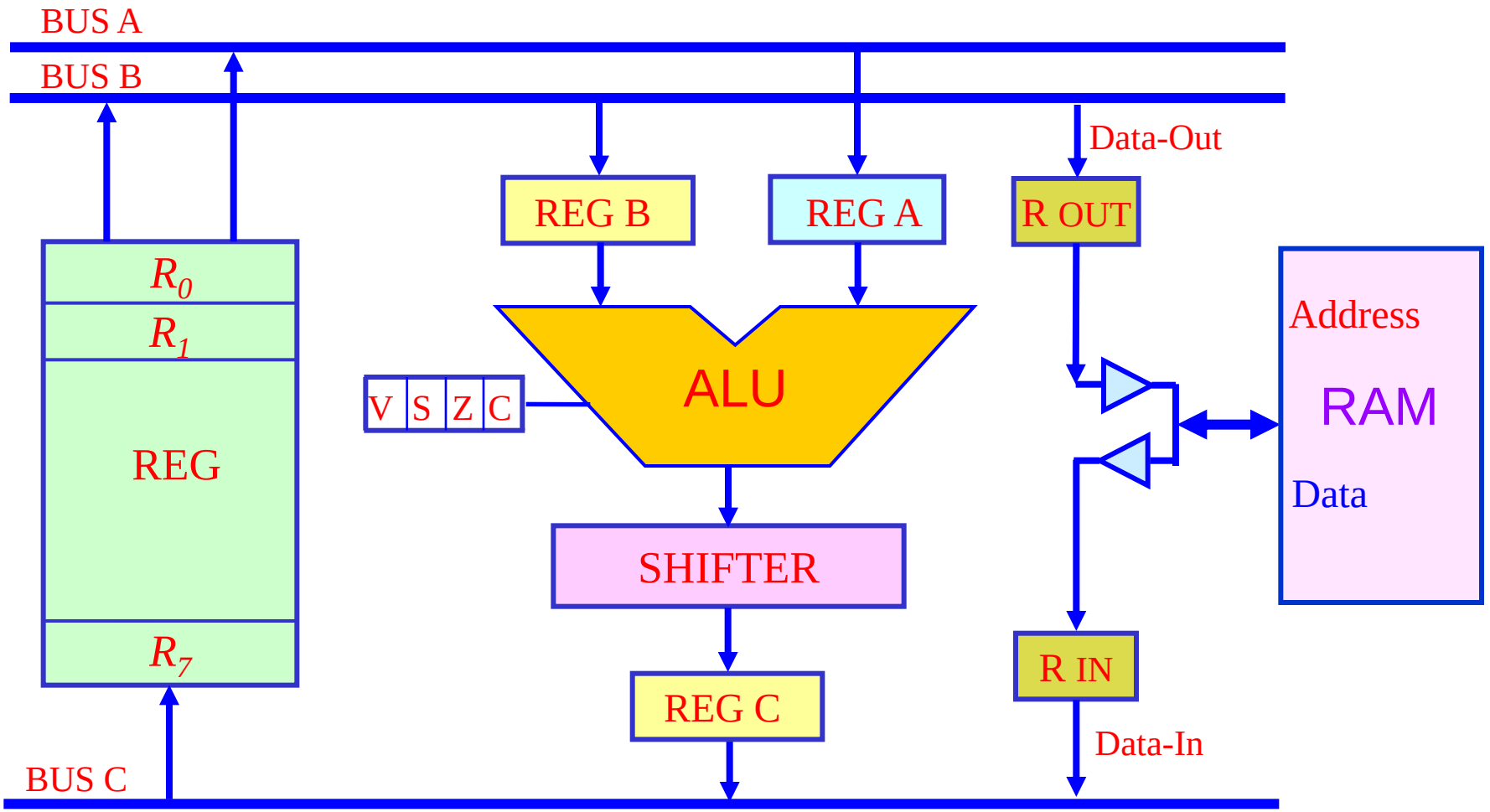
ASM diagram



Algoritmo → Procesador UV2009

$$R6, R7 \leftarrow M[500] * R4$$

Multiplication



Multiplicación

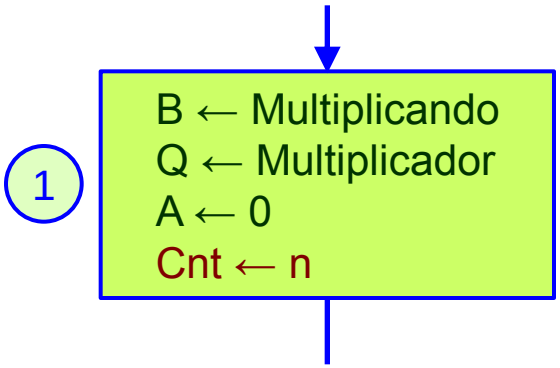
Universidad del Valle

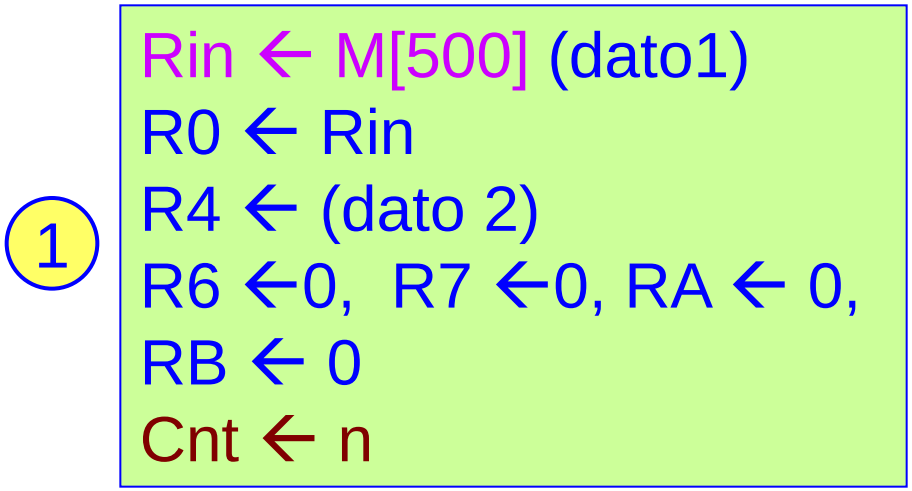
1. Cargar Datos

Registros a usar:

R0 = B (dato 1: multiplicando)
R4 = Q (dato 2: multiplicador)
R7 = Q (PL)
R6 = A (PH)

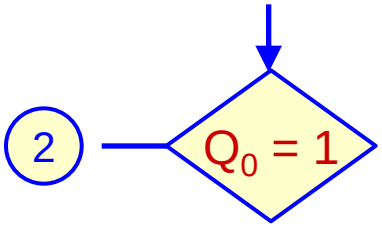
R1, R2 y R3 = Registros auxiliares





Multiplicación

2. Determinar Qo



2

RA ← R4, RB ← 0
RC ← Shr ([RA + RB], 0)
R1 ← RC
RA ← R1, RB ← 0
RC ← Shl ([RA + RB], 0)
R1 ← RC
RA ← R4, RB ←R1
RC ← R4 - R1

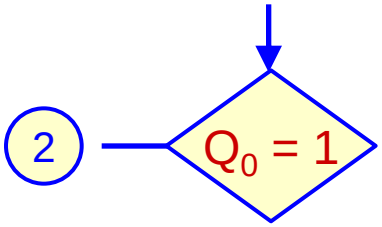
Entonces:

Si R4 - R1 = X, C=1
R4 - R1 = X, C=0

C=1, → Qo=0
C=0, → Qo=1

Multiplicación

2. Determinar Qo



2

RA ← R4, RB ← 0
RC ← Shr ([RA + RB], 0)
R1 ← RC
RA ← R1, RB ← 0
RC ← Shl ([RA + RB], 0)
R1 ← RC
RA ← R4, RB ←R1
RC ← R4 - R1

Entonces:

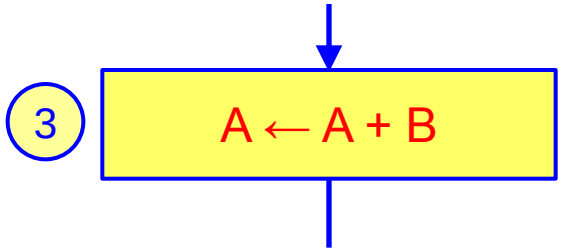
Si R4 - R1 = 0, Z=1
R4 - R1 = 1, Z=0

Z=1 → Qo=0
Z=0 → Qo=1

Multiplicación

Universidad del Valle

3. Sumar



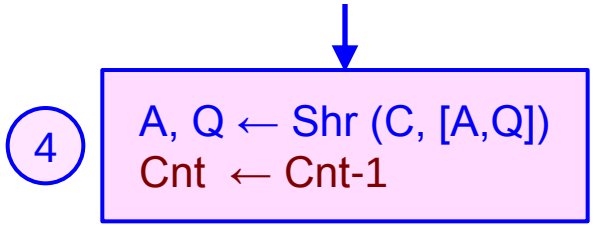
3

RA ← R6, RB ← R0
RC ← RA + RB
R6 ← RC

Multiplicación

Universidad del Valle

4. Rotar AQ con el carry [C]



1. Evaluar bit carry de operación: RC ← RA + RB

4

Si C=0:

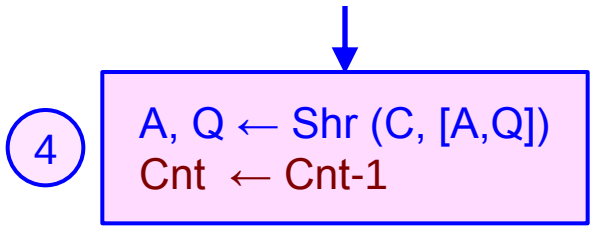
RA ← R6, RB ← 0
RC ← Shr ([RA + RB], 0)
R2 ← RC

Si C=1:

RA ← R6, RB ← 0
RC ← Shr ([RA + RB], 1)
R2 ← RC

Multiplicación

4. Rotar AQ con el carry [C]



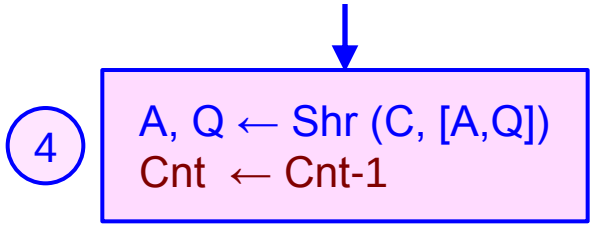
2. Evaluar bit R6[0]

4

RA ← R6, RB ← 0
RC ← Shr ([RA + RB], 0)
R3 ← RC
RA ← R3, RB ← 0
RC ← Shl ([RA + RB], 0)
R3 ← RC
RA ← R6, RB ← R3
RC ← R6 – R3

Multiplicación

4. Rotar AQ con el carry [C]



3. Evaluar bit zero de operación: RC ← R6 + R3

4

Si Z=0 (R6[0]=1):

RA ← R4, RB ← 0
RC ← Shr ([RA + RB], 1)
R4 ← RC, R7 ← RC
RA ← 0, RB ← 0
Cnt ← Cnt – 1

Si Z=1 (R6[0]=0):

RA ← R4, RB ← 0
RC ← Shr ([RA + RB], 0)
R4 ← RC, R7 ← RC
RA ← 0, RB ← 0
Cnt ← Cnt – 1

Calcular LSB. v1

Universidad del Valle

Calcular LSB [R0] = $R0_0$

1. $R0_{n-1} = 0$ Raux > R0

R0	111100
R_{aux}	111100
shr,0	011110 →
shl,1	111101 ←
R0	111100
- R_{aux}	111101
	111111

C = 1 → LSB = 0

2. $R0_{n-1} = 1$ Raux ≤ R0

R0	110111
R_{aux}	110111
shr,0	011011 →
shl,1	110111 ←
R0	110111
- R_{aux}	110111
	000000

C = 0 → LSB = 1

Calcular MSB: v1

Universidad del Valle

Calcular MSB [R0] = $R0_{n-1}$

1. $R0_{n-1} = 0$ Raux > R0

R0	011011
R_{aux}	100000
R0	011011
- R_{aux}	100000
	111011

C = 1 → MSB = 0

2. $R0_{n-1} = 1$ Raux ≤ R0

R0	110101
R_{aux}	100000
R0	110101
- R_{aux}	100000
	010101

C = 0 → MSB = 1

Calcular LSB: v2 (ALU)

Universidad del Valle

Calcular LSB [R0] = $R0_0$

1. $R0_{n-1} = 0$ Raux = 1

R0		101100
R_{aux}	And	000001
		000000

$Z = 0 \rightarrow LSB = 0$

2. $R0_{n-1} = 1$ Raux = 1

R0		100111
R_{aux}	And	000001
		000001

$Z = 1 \rightarrow LSB = 1$

Calcular MSB: v2

Universidad del Valle

Calcular MSB [R0] = $R0_{n-1}$

1. $R0_{n-1} = 0$ Raux = R0

R0		011011
R_{aux}	+	011011
R_{aux}		0 110110

C = 0 → MSB = 0

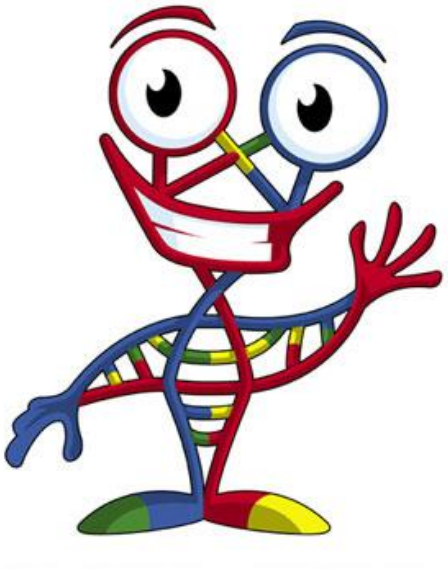
2. $R0_{n-1} = 1$ Raux ≤ R0

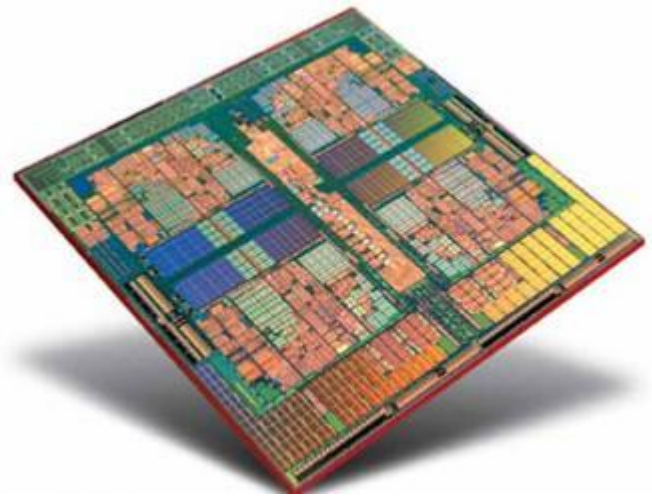
R0		110101
R_{aux}	+	110101
R_{aux}		1 101010

C = 1 → MSB = 1

R1 ← R0 + R0 ⇒ Shl(R0)

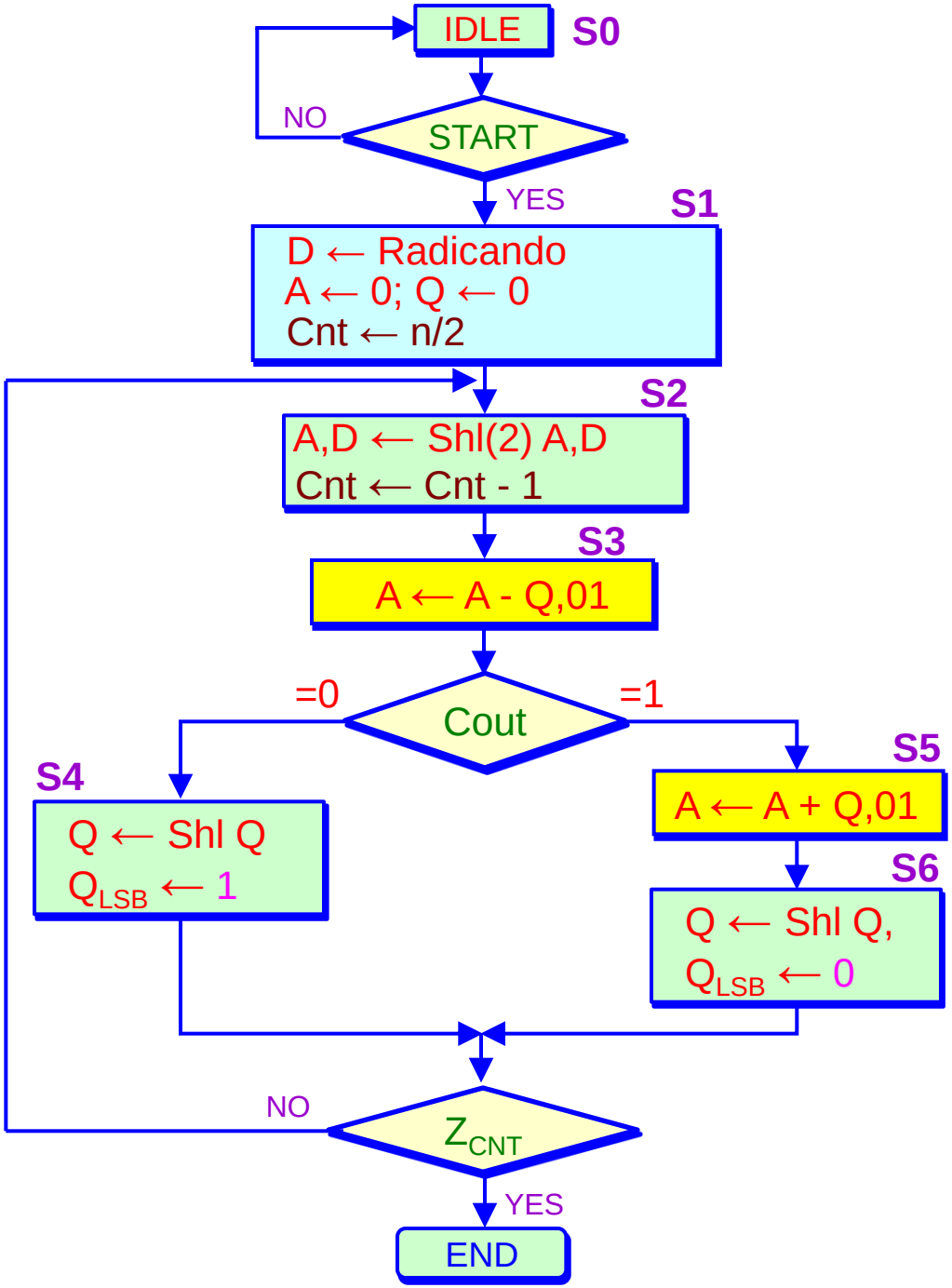
Square Root Design: √X





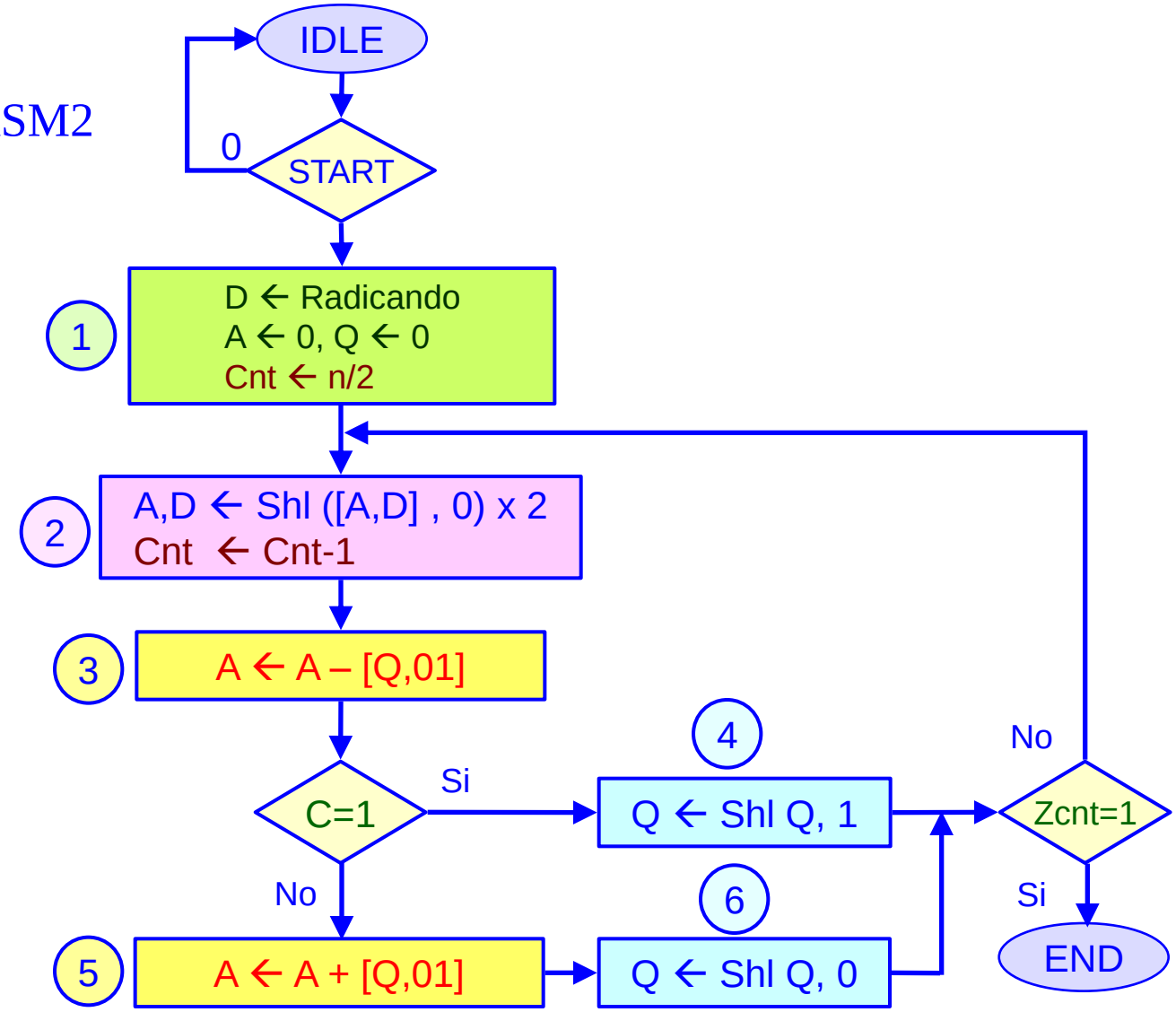
Raíz cuadrada

Algoritmo:
diagrama ASM1



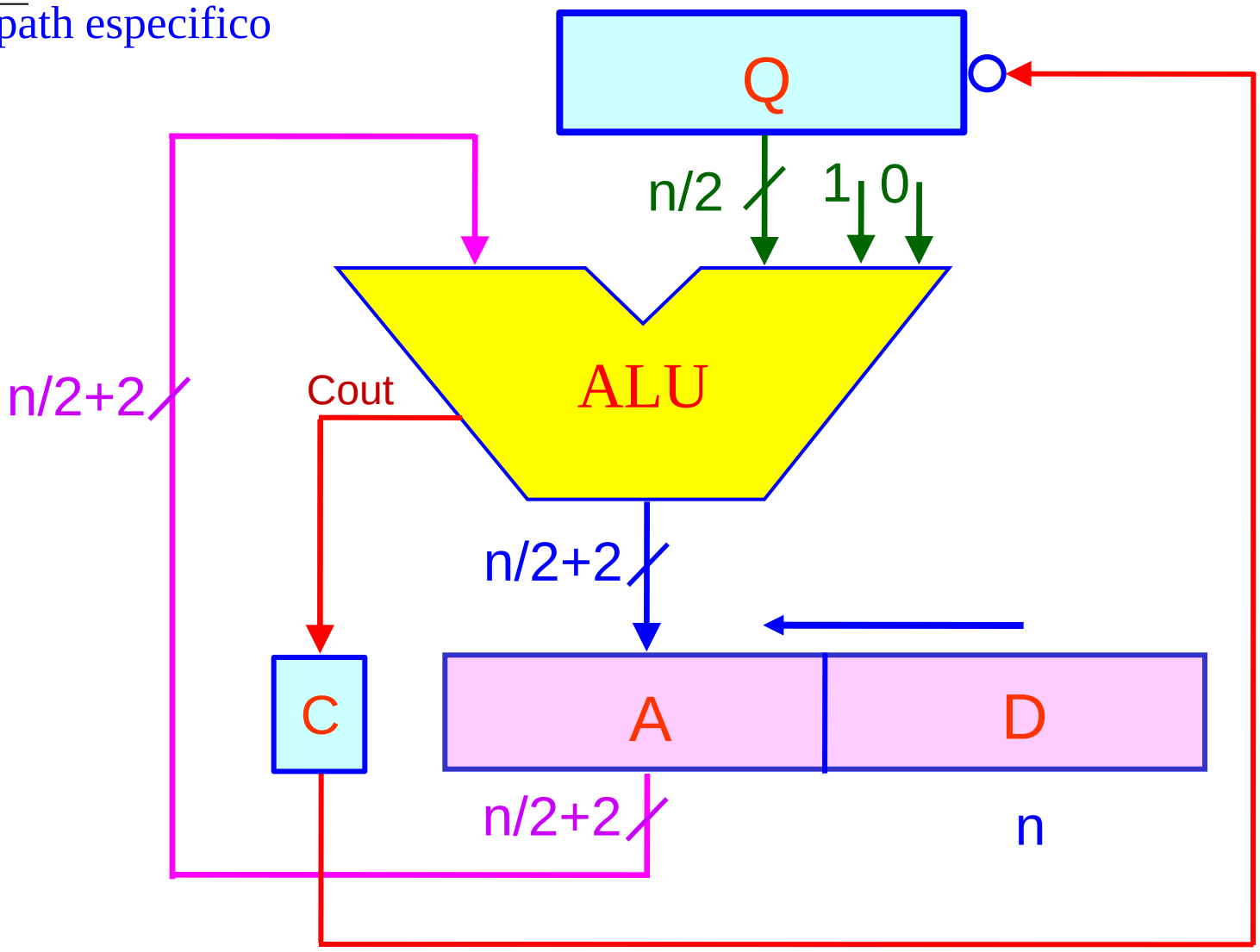
Raíz cuadrada

Algoritmo:
diagrama ASM2



Raíz cuadrada

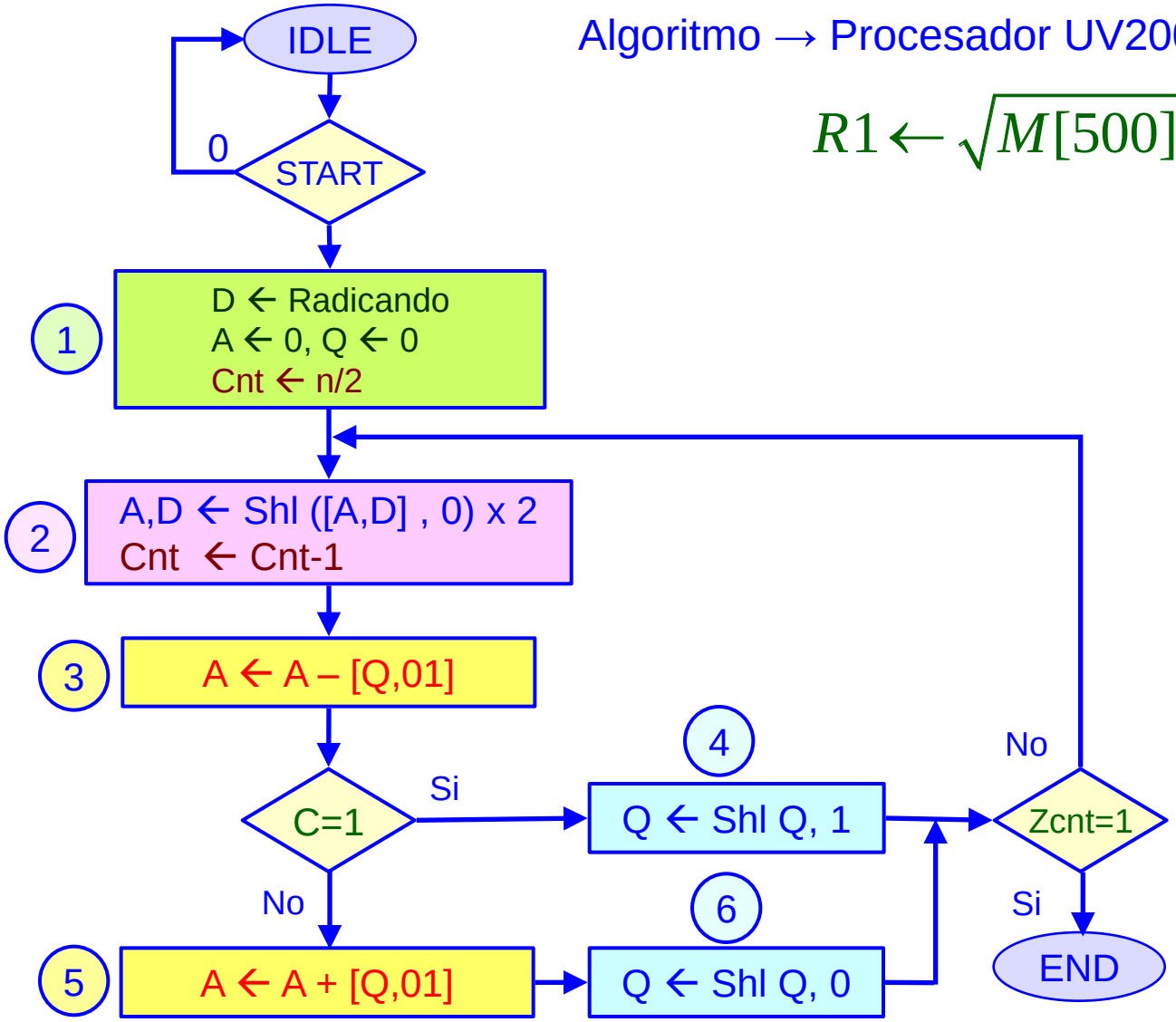
Datapath especifico



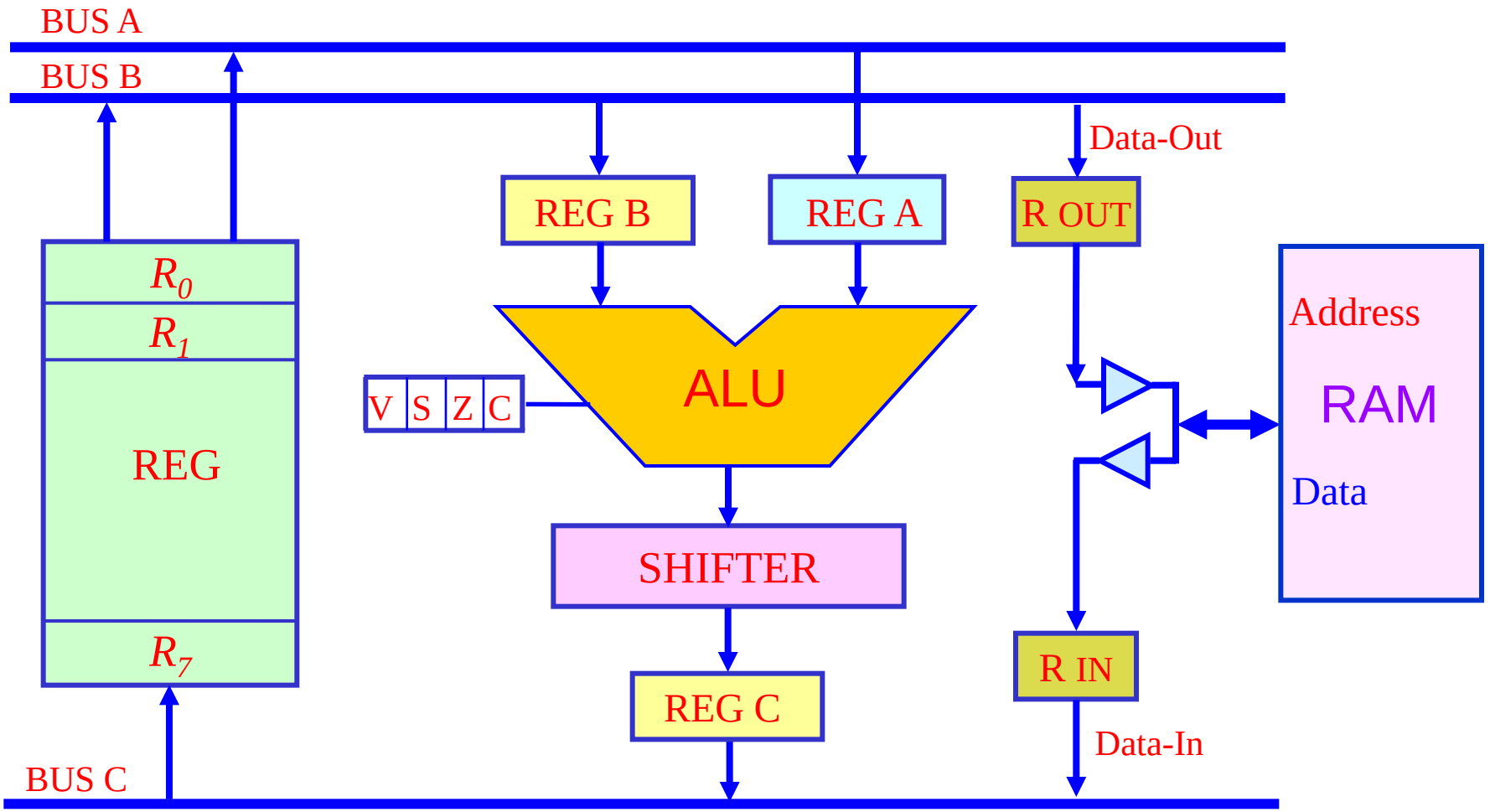
Raíz cuadrada

Algoritmo → Procesador UV2009

$$R1 \leftarrow \sqrt{M[500]}$$



Raíz cuadrada



Raíz cuadrada

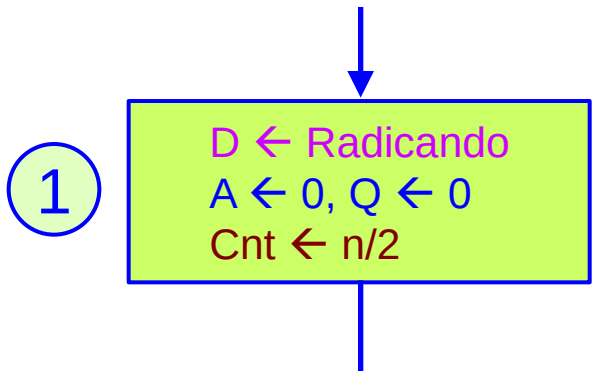
1. Cargar datos

Determinar los registros a usar:

R0 = D (dato1: radicando)
R1 = Q
R2 = A

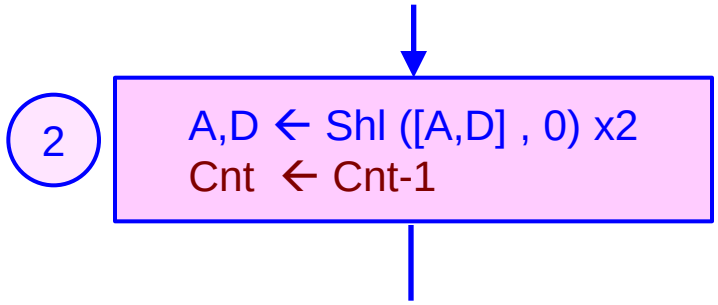
R3 y R6 = Registros auxiliares

$$R1 \leftarrow \sqrt{M[500]}$$



Raíz cuadrada

2. Doble desplazamiento del registro A,D



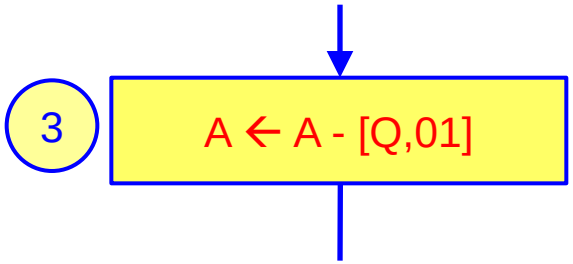
R0 = D = 0110
R1 = Q
R2 = A
R3 y R6 = Reg Auxiliar

2

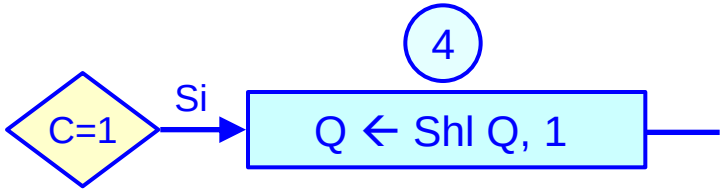
RA ← R0, RB ← 0
RC ← Shr (RA, 0)
R3 ← RC
RA ← R3, RB ← 0
RC ← Shr (RA, 0)
R3 ← RC
RA ← R2, RB ← R2
RC ← Shl ([RA +RB], 0)

R2 ← RC
RA ← R2, RB ← R3
RC ← RA + RB
R2 ← RC
RA ← R0, RB ← R0
RC ← Shl ([RA +RB], 0)
R0 ← RC
Cnt ← Cnt – 1

Raíz cuadrada



Si C = 1 (A ≥ [Q,01])



3

RA ← R1, RB ← R1
RC ← Shl ([RA + RB], 1)
R6 ← RC
RA ← R2, RB ← R6
RC ← RA – RB
R3 ← RC

4

RA ← R1, RB ← 0
RC ← Shl ([RA + RB], 1)
R1 ← RC

Raíz cuadrada

Si C = 0 (A < [Q,01])

5

RA ← R2, RB ← R1
RC ← RA + RB
R2 ← RC

6

RA ← R1, RB ← 0
RC ← Shl ([RA + RB], 0)
R1 ← RC

Raíz cuadrada

Algoritmo → Procesador UV2009

$$R1 \leftarrow \sqrt{M[500]}$$