Sergio David Marquez 1710075

26

C = 3.53

Paquetes

Sensar LSB Paquete 1

1. RA ← Rx ; RB ← 1 ; rRx, WRA, WRB
2. RC ← RA AND RB ; WALU(11)
3. RAux ← RC ; rRC; WRAux
4. RA ← RAux RB ← 1 ; WRA; rRAux; WRB
5. Rc ← RA − RB ; FC ← Cout; ALU(10) ; ELK(FC)

Paquete 2 sensar MSB

1. RA ← Rx ; RB ← Rx ; rRx WRA; WRB
2. Rc ← RA+RB; FC ← cout ; ALU(01); EN(FC)

Paquete 3 desplaza la derecha con 0 Shifter

1. RA ← Rx ; RB ← 0 ; rRx, WRA, WRB
2. RC ← (RA + RB) Shr ; Shifter(010)
3. Rx ← RC ; WRx, rRC

Paquete 4 desplaza derecha con 1

1. RA ← Rx ; RB ← 0 ; rRx, WRA, WRB
2. RC ← Shr(RA+RB, 1) ; Shifter(011)
3. Rx ← RC ; WRx, rRC

Paquete 5 desplaza Izq con 0

1. RA ← Rx ; RB ← 0; WRA, rRx, WRB
2. RC ← Shl(RA+RB, 0); Shifter(000)
3. Rx ← RC ; RX2 ← C ; WRx, rRC

OK solo registro

Paquete 6

Sumar Rx + Ry sensando cout

RA ← Rx RB ← Ry ; rRx, WRA, WRB, rRy

RC ← RA + RB, FC ← cout; ALU(01), EN(FC)

Rx ← Rc ; rRc, WRx

Paquete 7 restar Rx − Ry sensando cout

1. RA ← Rx , RB ← Ry ; rRx, WRA, rRy, WRB
2. RC ← (RA − RB), FC ← cout ; ALU(10); EN(FC)
3. RA ← RC ; rRC, WRA

Paquete 8 desplaza Izq con 1

1. RA ← Rx; RB ← 0 ; WRA, rRx, WRB
2. RC ← Shl(RA+RB, 1) ; Shifter(001)
3. Rx ← RC ; RX2 ← C ; rRC, WRB,

OK solo registro

Paquete 9 transferir Rx a Ry

1. RA ← Rx ; RB ← 0 ; rRx; WRA, WRB
2. RC ← Sh(RA) ; Shifter(100), ALU(00)
3. Ry ← RC ; rRc, WRy

Paquete 10 Desplazamiento aritmetico derecha

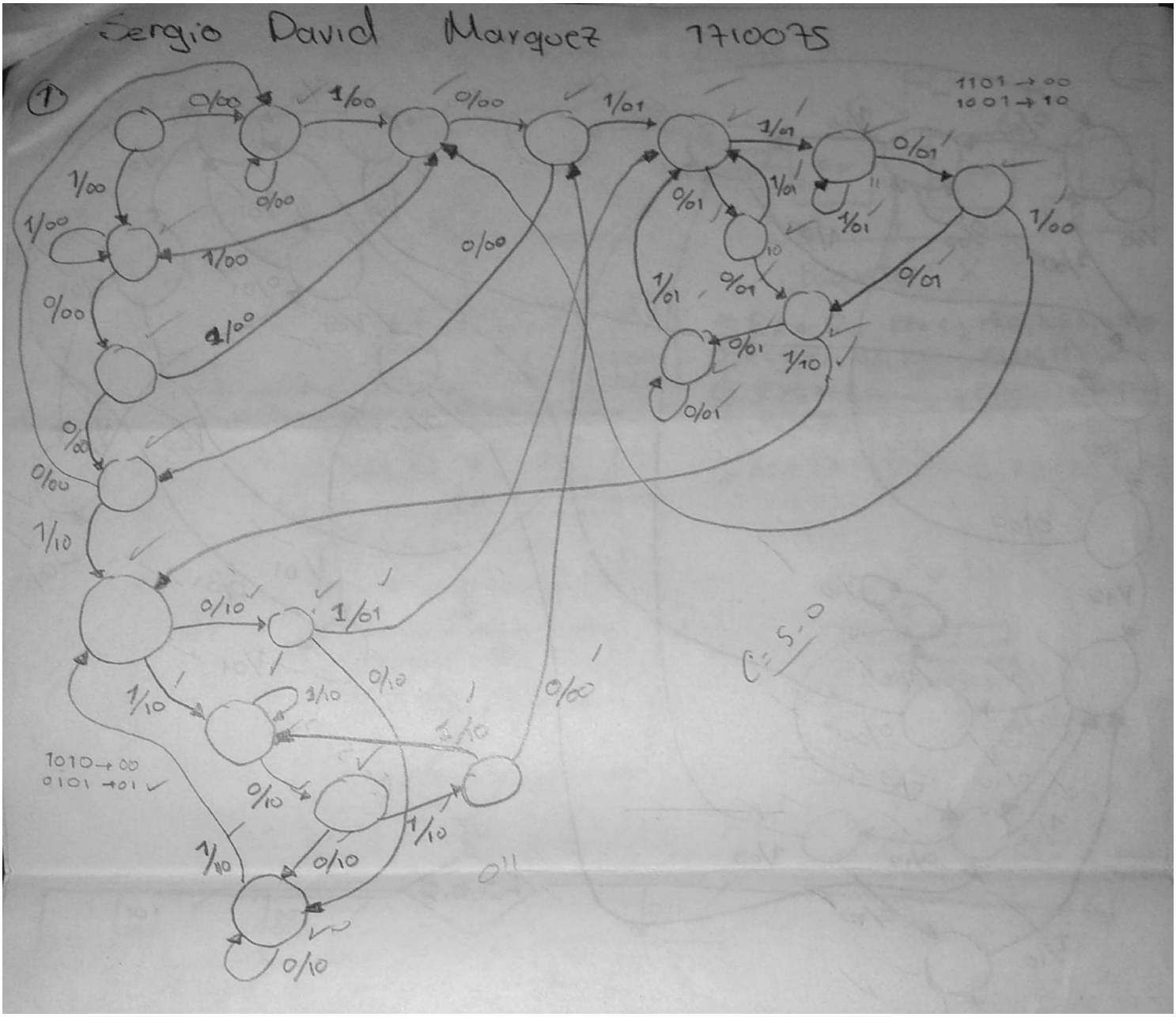
1. RA ← Rx RB ← 0 ; rRx, WRA; WRB
2. Rc ← ShrA(RA) ; Shifter(101), ALU(00)
3. Rx ← RC ; rRc; WRx

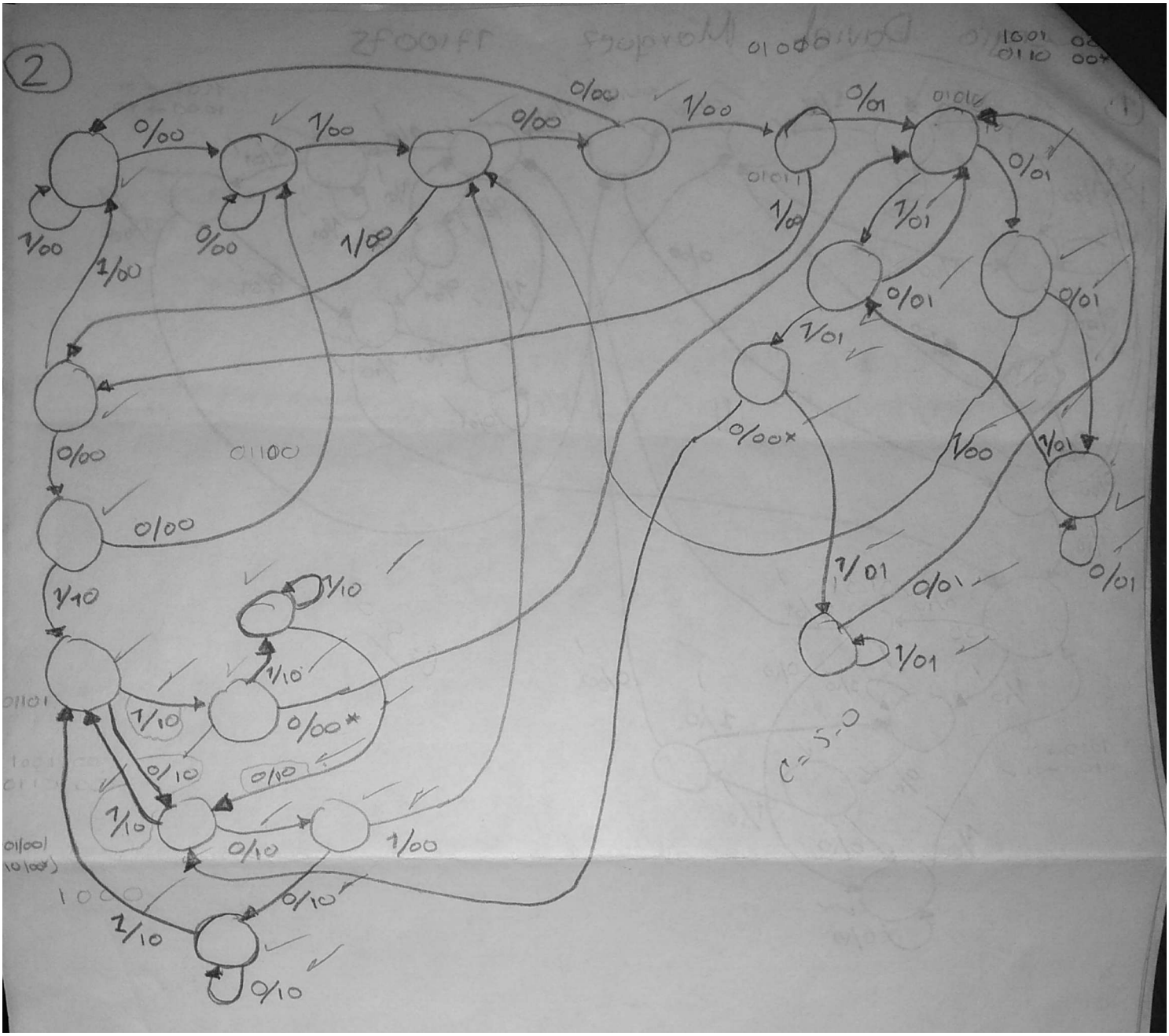
15.0
25.0
J45
445
50.0 bbe

Paquete 11 swap a Rx

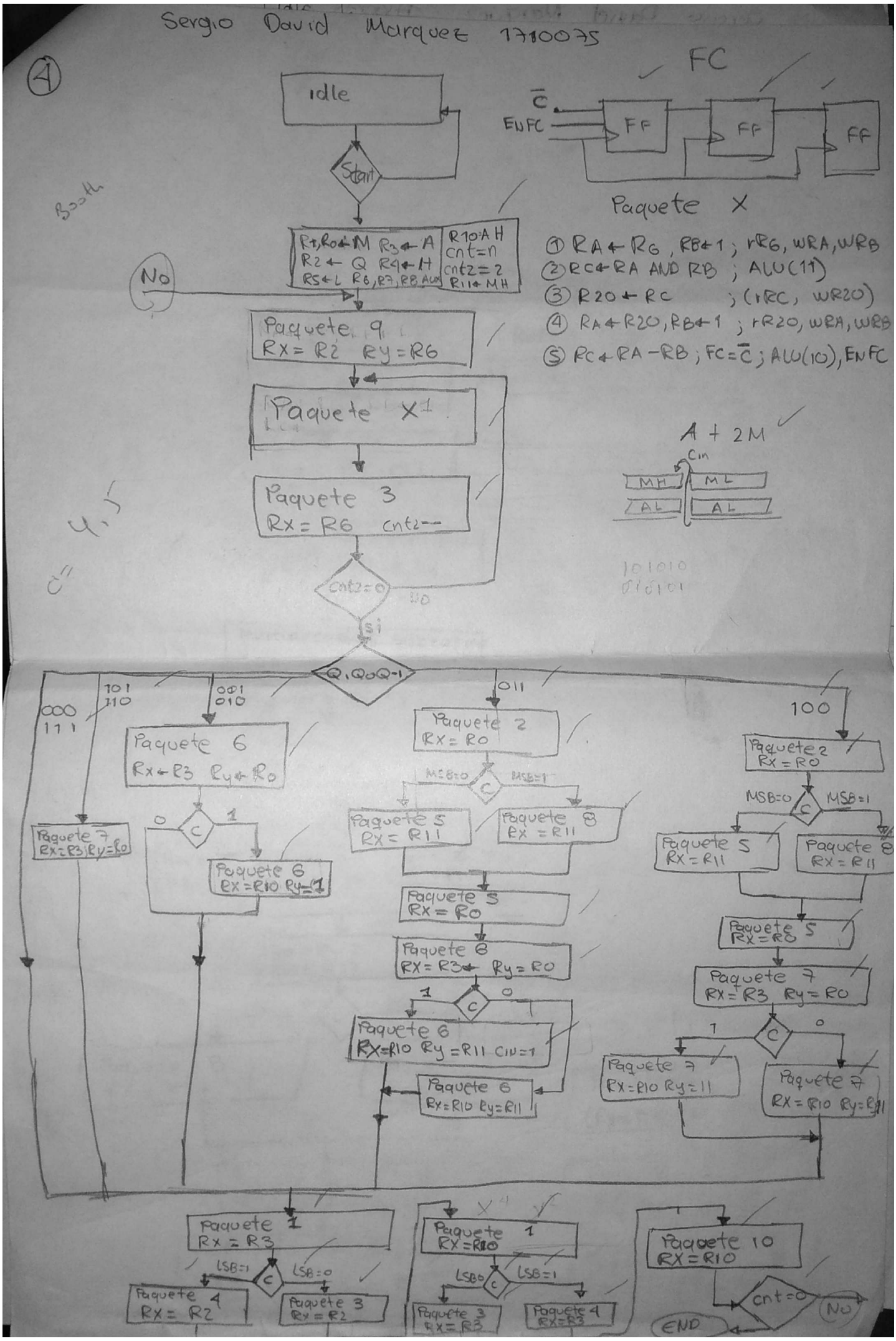
1. RD ← Rx ; WRD, rRx
2. RE ← R(Swap(RD)) ; ASwap
3. Rx ← RE ; rRE, WRx

Sergio David Marquez 7710075




2
0/00
1/00
0/00
1/00
0/00
1/00
0/01
0/01
1/00
1/00
1/00
0/01
1/01
0/01
0/01
0/00
0/00
1/01
0/00*
1/00
1/01
0/01
0/01
1/01
1/10
1/10
1/10
1/10
0/00*
1/10
0/10
0/10
1/10
0/10
1/00
0/10
1/10
0/10
C=5.0

Sergio David Marquez 1710075

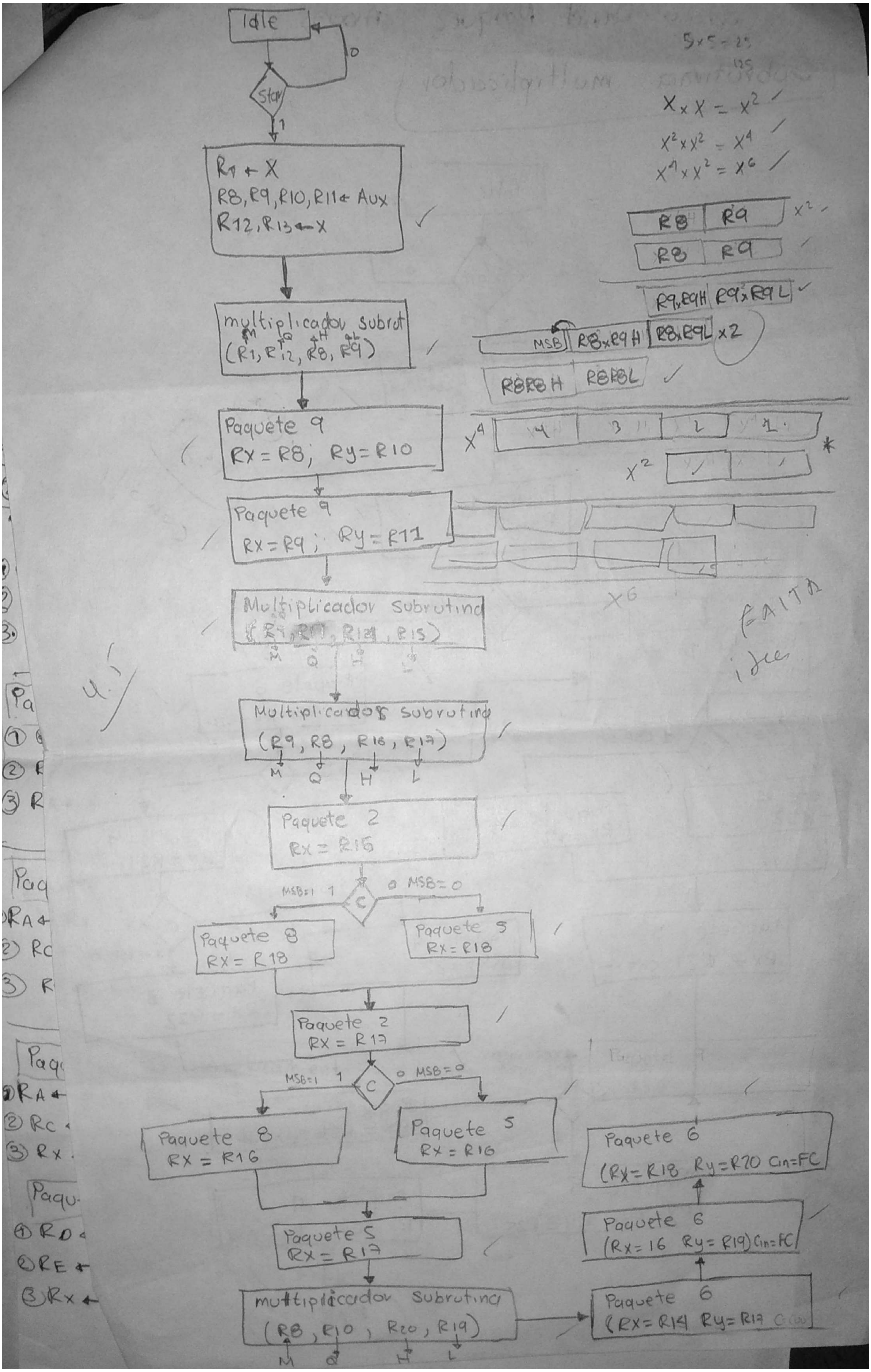


FC

Paquete X

① RA ← R6 , RB ← 1 ; rR6, WRA, WRB
② RC ← RA AND RB ; ALU(11)
③ R20 ← RC ; (rRC, WR20)
④ RA ← R20, RB ← 1 ; rR20, WRA, WRB
⑤ RC ← RA − RB ; FC = C̄ ; ALU(10), EnFC

A + 2M



Idle
Start
0
1
R1 ← X
R8, R9, R10, R11 ← Aux
R12, R13 ← X
multiplicador subrut
(R1, R12, R8, R9)
Paquete 9
RX = R8; Ry = R10
Paquete 9
RX = R9; Ry = R11
Multiplicador subrutina
(R9, R11, R14, R15)
M Q H L
Multiplicador subrutina
(R9, R8, R16, R17)
M Q H L
Paquete 2
Rx = R16
MSB=1 1
C
0 MSB=0
Paquete 8
RX = R18
Paquete 5
RX = R18
Paquete 2
RX = R17
MSB=1 1
C
0 MSB=0
Paquete 8
RX = R16
Paquete 5
RX = R16
Paquete 5
RX = R17
multiplicador subrutina
(R8, R10, R20, R19)
M Q H L
Paquete 6
(RX = R14 Ry = R17
Paquete 6
(Rx = 16 Ry = R19) Cin=FC
Paquete 6
(Rx = R18 Ry = R20 Cin=FC
5x5 = 25
125
X x X = X^2
X^2 x X^2 = X^4
X^4 x X^2 = X^6
R8 R9 X^2
R8 R9
R9xR9H R9xR9L
MSB R8xR9H R8xR9L x2
R8R8H R8R8L
X^4 4 3 2 1
X^2
X^6
FALTA
idea

Sergio David Marquez A00975

Subrutina multiplicador