ALU

Brebu Iasmin Marian

Input:

X(w biti), Y(w biti), OP(4 biti), clk, rst_b

w de forma 2^n unde n numar natural, parametru este si numarul de biti din counter care trebuie sa fie cnt7size = log2(w).

OP reprezinta operatia de efectuat:

1. X+Y
2. X-Y
3. X*Y
4. X/Y
5. x<<1
6. x>>1
7. X AND Y (logic ca in C)
8. X OR Y (logic ca in C)
9. X XOR Y (logic ca in C)
10. X bAND Y (x[i] & y[i])
11. X bOR Y (x[i] | y[i])
12. X bXOR Y (x[i] ^ y[i])

Cand s-a identificat o operatie valida, aceasta va fi efectuata, semnalul de OP fiind ignorat pana la terminarea operatiei.

Toate operatiile suporta numere negative in C2.

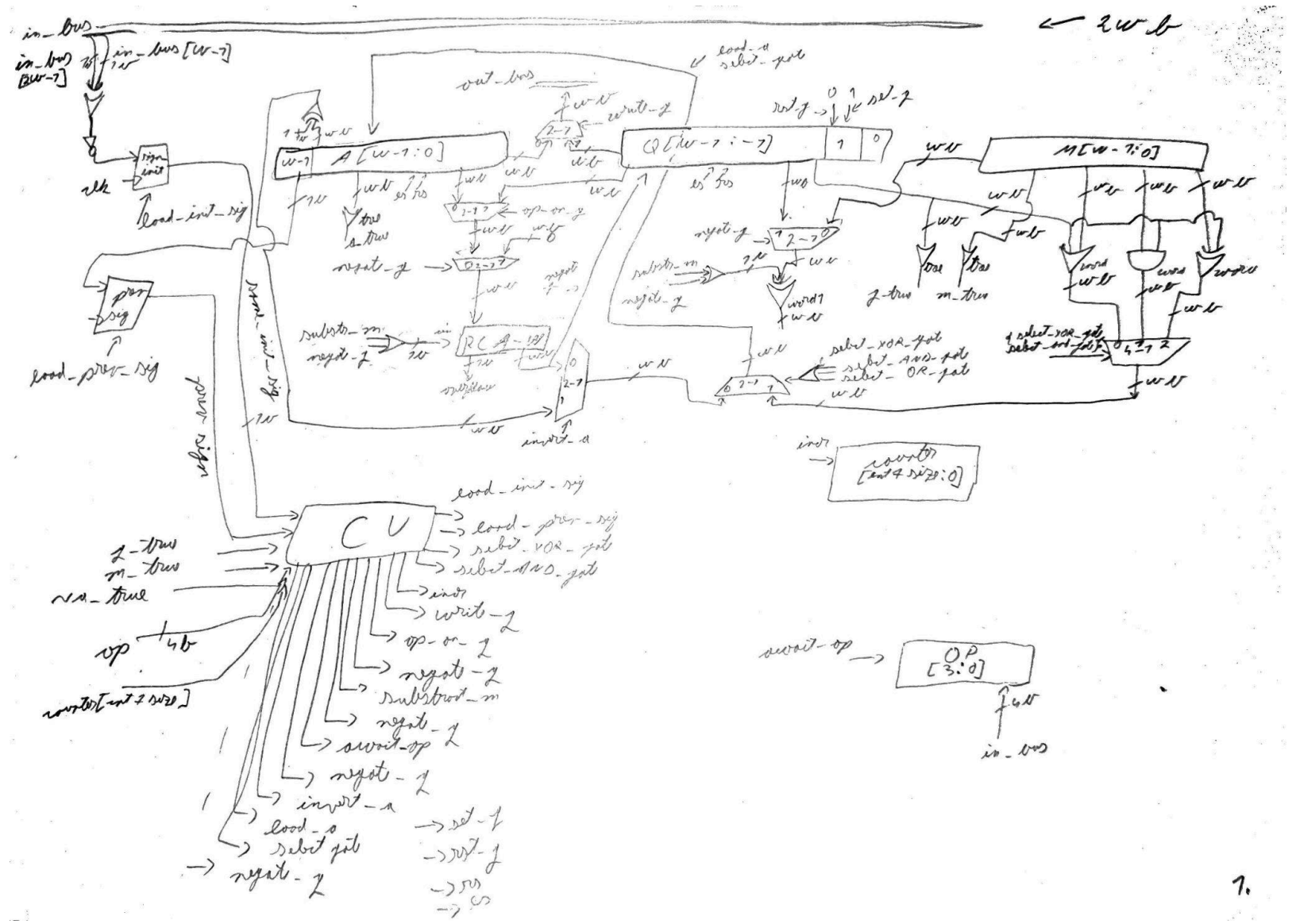
Output:

Z(w biti), status

Cand status=1 rezultatul operatiei se afla in Z.

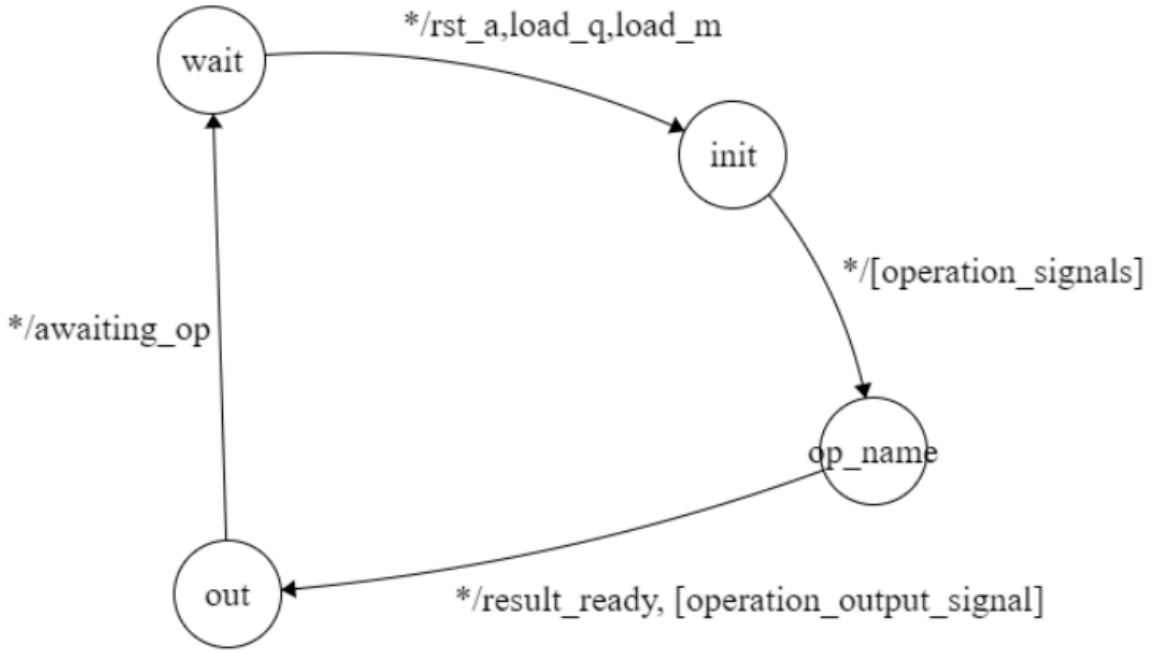
La inmultire/impartire status va fi 1 doua tacuri de clk, in fiecare tac in Z aflanduse o jumatate din rezultat. Pentru primul clk: la inmultire va fi jumatatea cea mai semnificativa din rezultat, iar la impartire va fi catul. Pentrul al doilea clk: la inmultire jumatatea cea mai putin semnificativa din rezultat, iar la impartire restul.

Pentru AND, OR, si XOR, Z va avea toti bitii 1 daca rezultatul este adevarat, si toti bitii 0 daca rezultatul este fals.



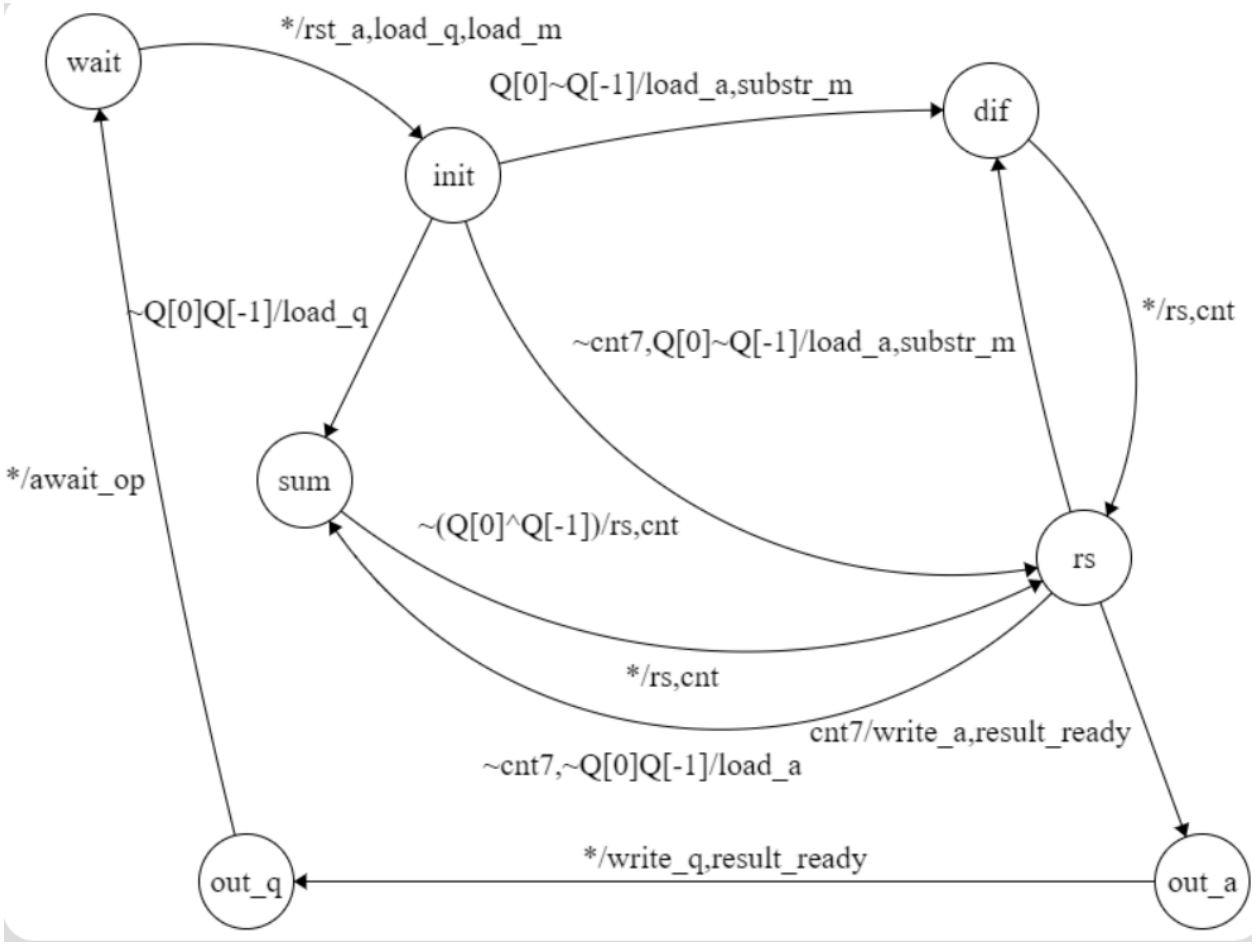
States

Sum, dif, rs, ls, AND, OR, XOR, bAND, bOR, bXOR

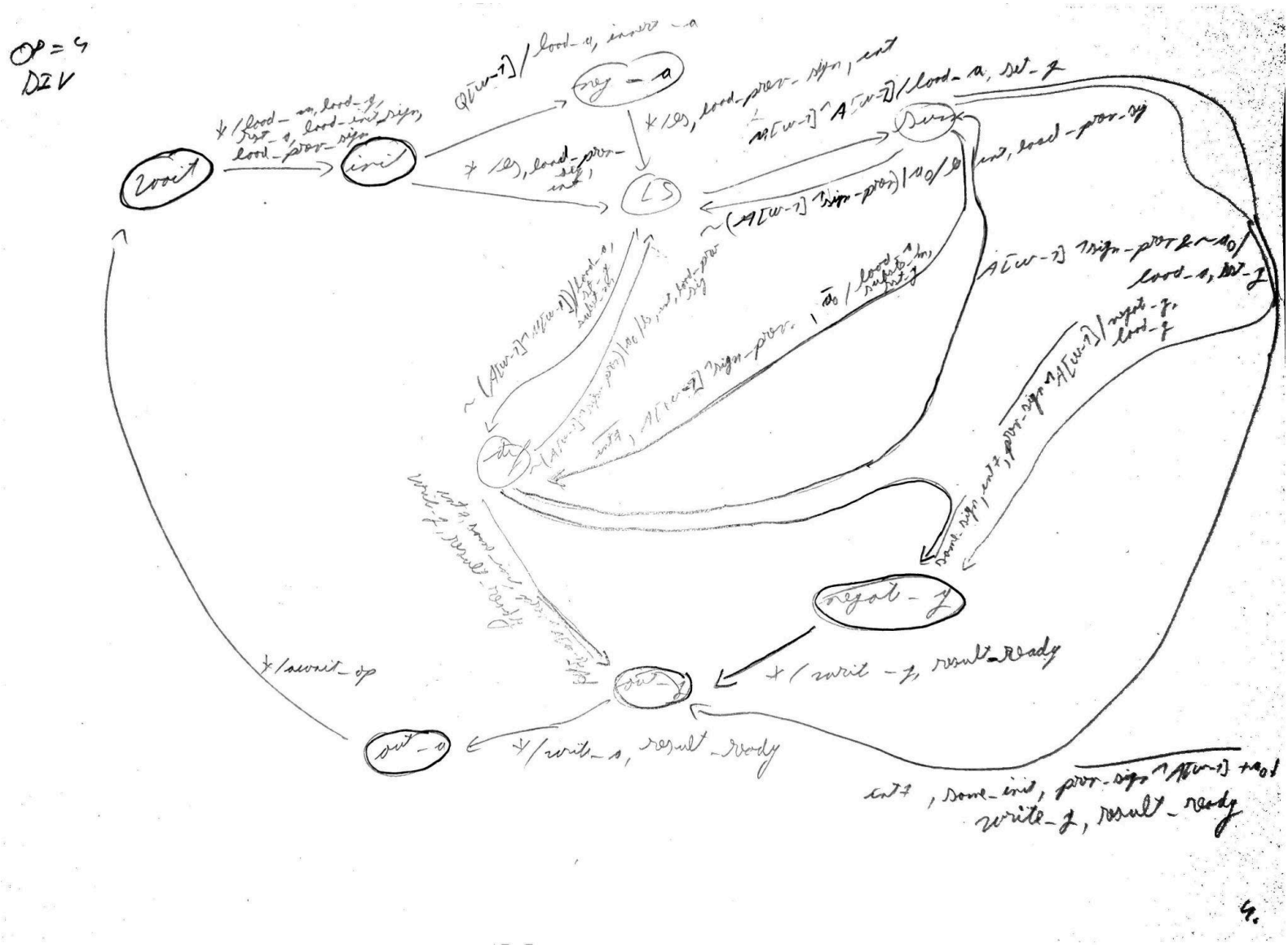


	[operation_signals]	[operation_output_signal]
Sum:	load_a	write_a
Dif:	load_a, substract_m	write_a
LS:	ls	write_q
RS:	rs	write_q
AND:	a_and_m & negate_a, a_and_m & load_a	write_a
OR:	a_or_m & negate_a, a_or_m & load_a	write_a
XOR:	a_xor_m & negate_a, a_xor_m & load_a	write_a
bAND:	select_and_gate	write_a
bOR:	select_or_gate	write_a
bXOR:	select_xor_gate	write_a

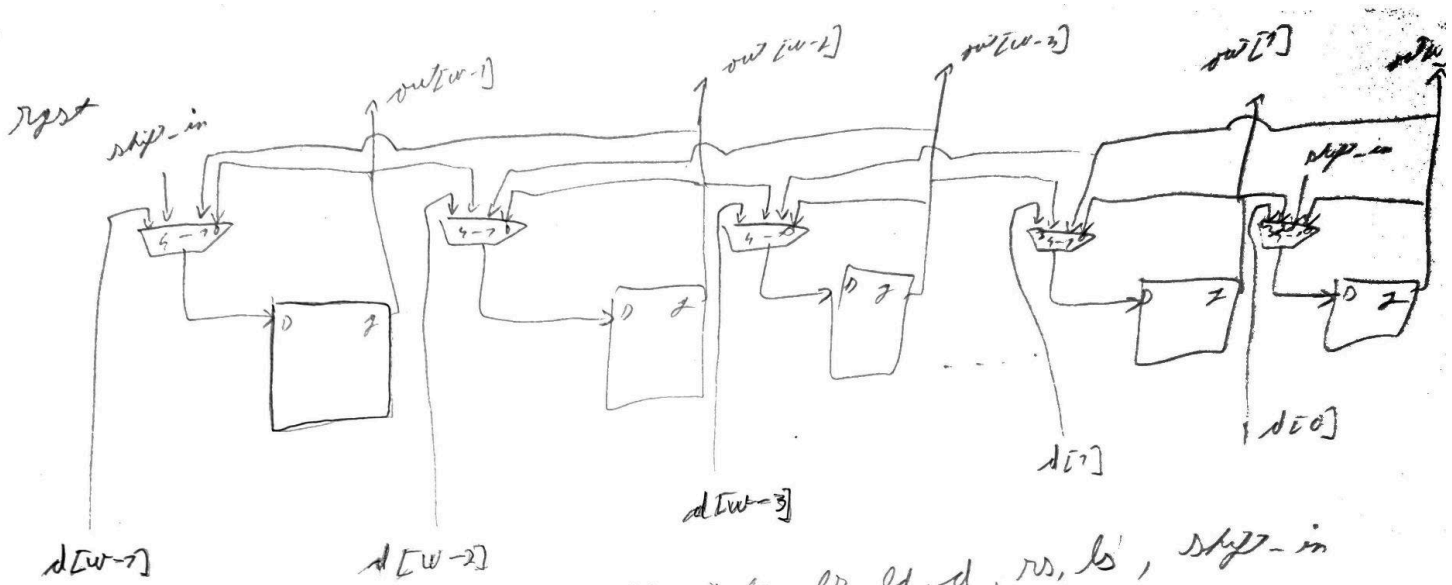
Multiplication



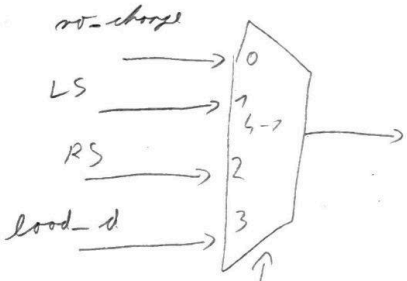
Division



LEFT RIGHT SHIFT REGISTER

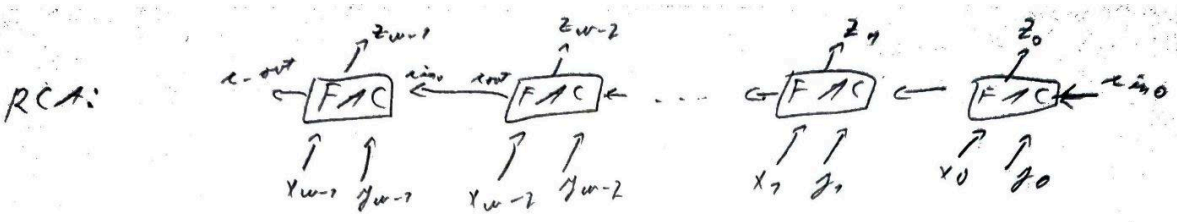


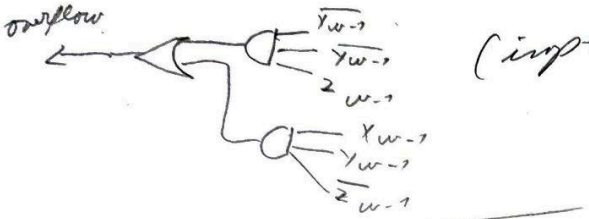
Input: clk, rst_L, clr, ld, d, rs, ls, shift_in
Output: Q
Behaviour: rs ⇒ right shift & pad with shift_in
ls ⇒ left shift
rs & ls ⇒ nothing
ld ⇒ load d in Q

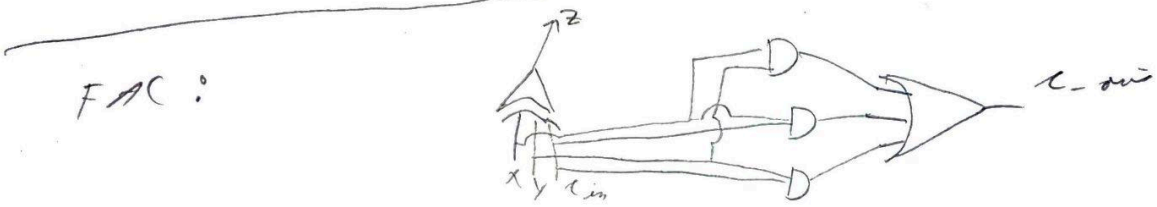


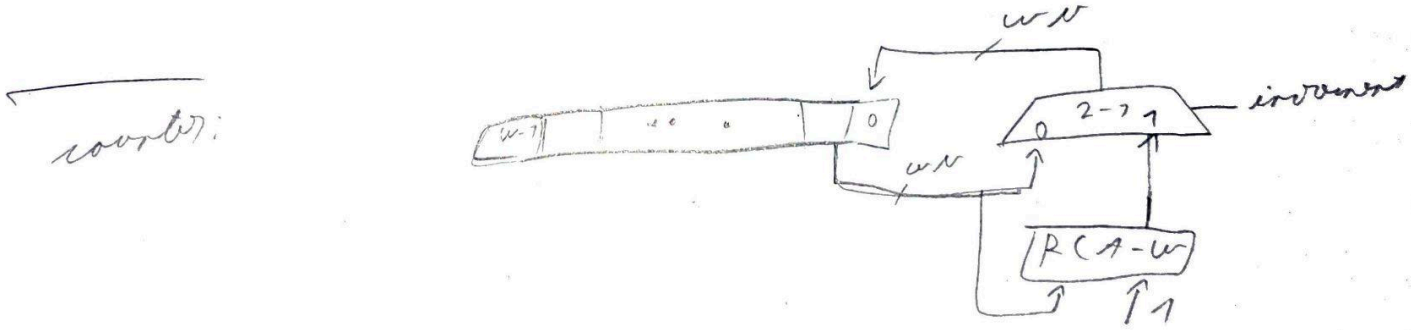
$\overline{ld} \cdot \overline{ls} \cdot rs + ld \cdot \overline{ls} \cdot rs$, $\overline{ld} \cdot ls \cdot \overline{rs} + ld \cdot \overline{ls} \cdot \overline{rs}$

RCA and COUNTER



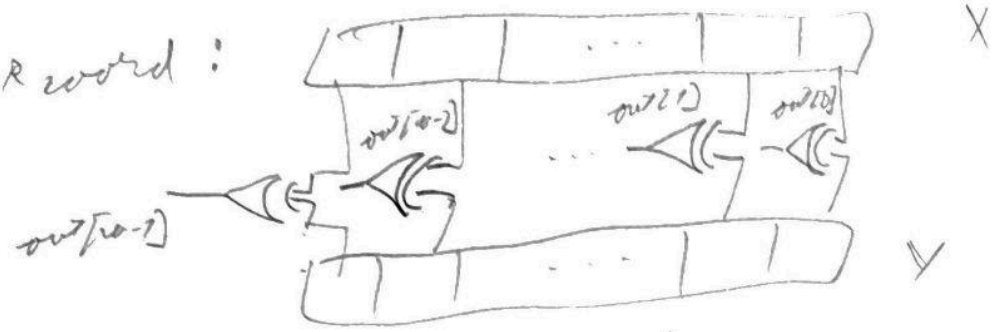






Word Gates & Tree

XOR word :



OR word, AND word idem

XOR word1 : idem dar Y = 111...111

OR tree :