

Lucrare:

Interfatarea unui unei imprimante Printronix Model 300, folosind o interfata standard Centronics, la un sistem Motorola cu procesor MC68000.

In cadrul sistemului pentru a se putea realiza comunicatia intre imprimanta si procesorul Motorola 68000, s-a folosit chip-ul MC68230 Paralel Interface Timer.

Descrierea modului de functionare al MC68230:

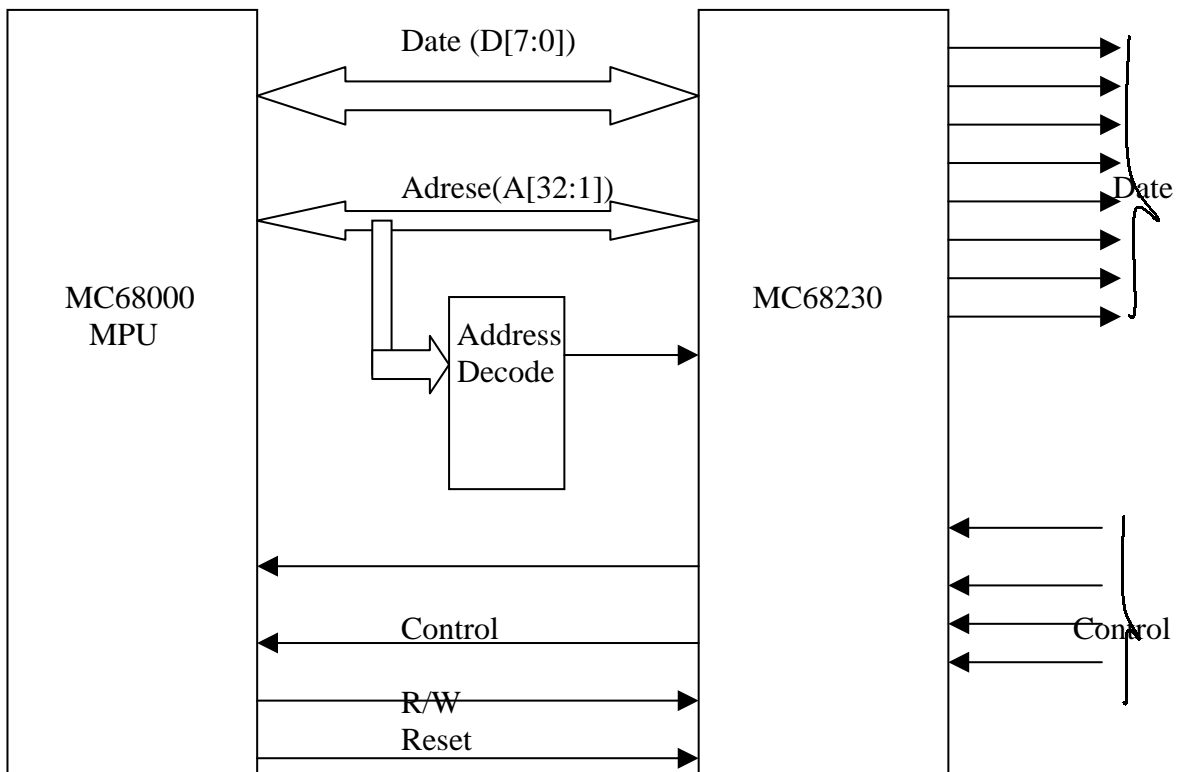
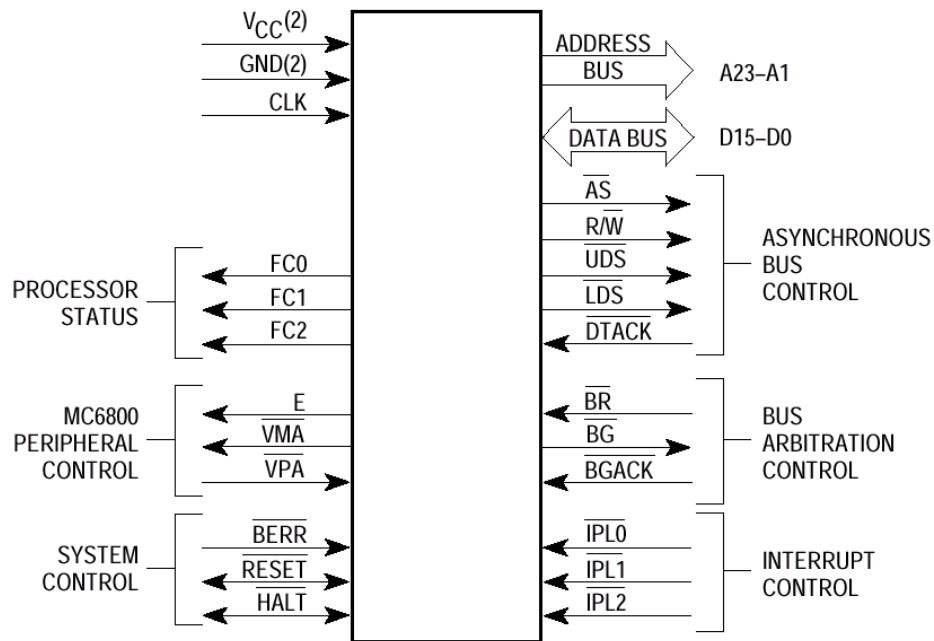
- permite transferul de date prin doua semnale de “handshake”, DATA STROBE si ACKNLG
- mai exista si semnale aditionale ce vin de la printer pentru a semnaliza starea acestuia la un moment dat: BUSY, PE (“printer error”, de exemplu “out of paper”) si SLCT (select)
- controlerul trimite date pe liniile PA0 – PA6 si H2 folosit pentru “data strobe”; cand printerul primeste date pe aceste linii trimite inapoi la controler un ACKNLG pe pinul H1 al controlerului MC68230, acest lucru indica faptul ca printerul este gata sa primeasca urmatorul caracter
- daca printerul nu poate fi accesat din diferite motive, intoarce 1 pe linia BUSY ce corespunde pinului PA7 din controlerul de imprimanta; in acest caz nu se trimit date spre printer pana cand nu se pozitioneaza ACKLNG pe 1

Descrierea semnalelor si a modului de functionare pentru microprocesorul MC68000:

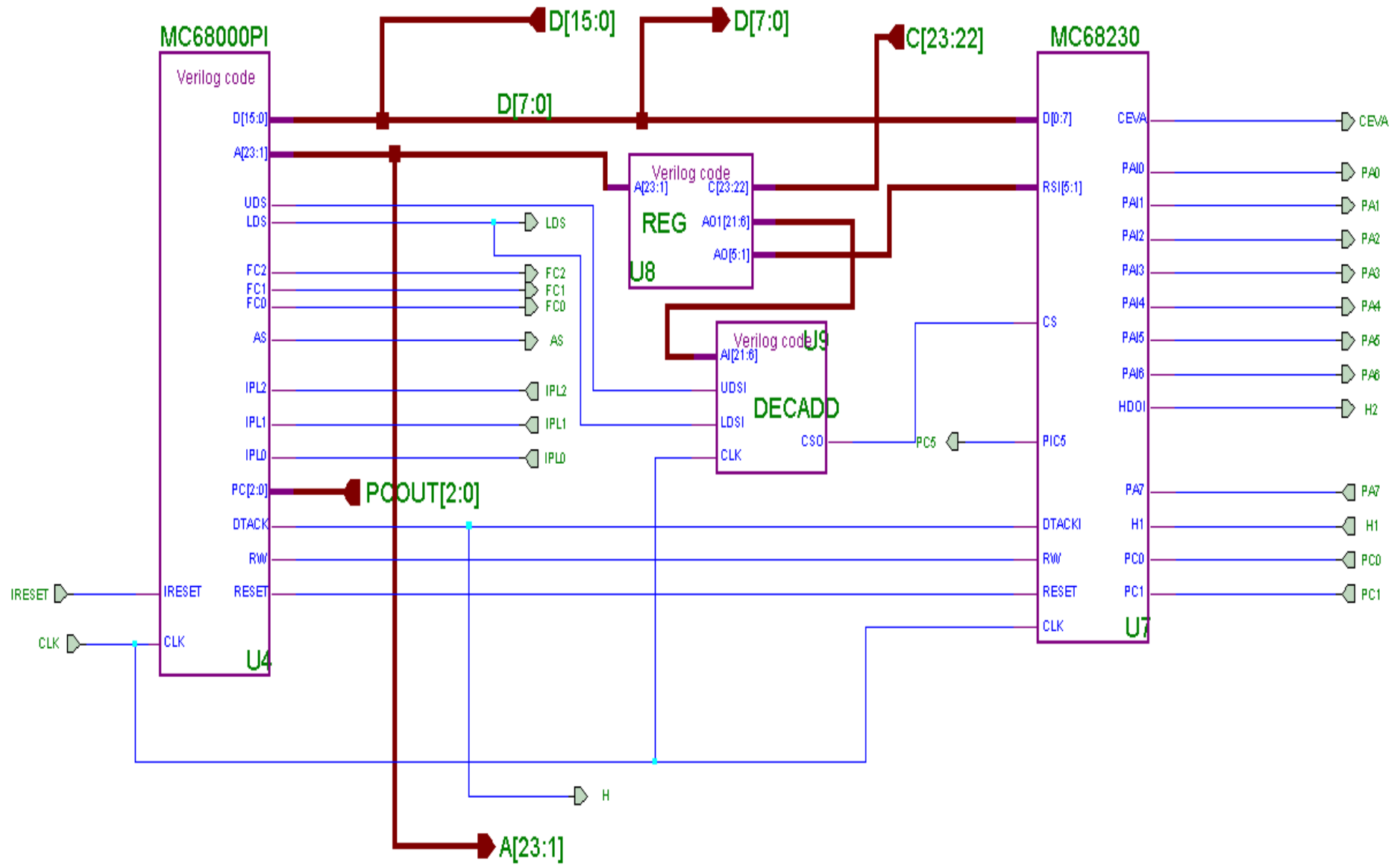
- Semnale de date -> magistrala pe 16 biti (D0 – D15), din care pentru lucrul cu imprimanta vom folosi doar bitii D0 – D7
- semnale de adresa A1 – A23 folosite pentru identificarea dispozitivului
- R/W – read/write control, semnal folosit pentru a specifica tipul de operatie efectuat
- UDS, LDS – upper/lower data stobes, pozitionate pe 0, respectiv 1 pentru selectarea D0 – D7 in lucrul cu imprimanta
- DTACK – data ack, folosit pentru a instiinta daca transferul a fost efectuat corect
- RESET – semnal ce poate reseta sistemul, sau doar anumite periferice

Vom prezenta mai jos schema de principiu a procesorului MC68000 si a schemei de interconectare la imprimanta, iar apoi schema detaliata, codul verilog pentru modulele M68000 si MC68230, si simularea schemei in Xilinx.

Schema generala M68000:



snap.bmp (1206x556x24b bmp)



```

1: module MC68000 (CLK, D, A, LDS, FC2, FC1, FC0, AS, UDS, RW, RESET, DTACK, IPL2, IPL1, IPL0, IRE
2:
3: input CLK ;
4: input DTACK ;
5: input IPL0 ;
6: input IPL1 ;
7: input IPL2 ;
8: input IRESET ;
9:
10: output [2:0] PC ;
11: output [15:0] D ;
12: output [23:1] A ;
13: output LDS ;
14: output FC2 ;
15: output FC1 ;
16: output FC0 ;
17: output AS ;
18: output UDS ;
19: output RW ;
20: output RESET ;
21:
22: reg [7:0] Mem [7:0];
23: reg [2:0] PC;
24: reg LDS, UDS, RW, RESET, FC2, FC1, FC0, AS;
25: reg [15:0] D;
26: reg [23:1] A;
27:
28: // add your declarations here
29:
30: always @(negedge CLK)
31: begin
32: #1 Mem[0] = 8'b000000010;
33: #1 Mem[1] = 8'b000000110;
34: #1 Mem[2] = 8'b00001110;
35: #1 Mem[3] = 8'b00011110;
36: #1 Mem[4] = 8'b00111110;
37: #1 Mem[5] = 8'b01111110;
38: #1 Mem[6] = 8'b11111110;
39: #1 Mem[7] = 8'b11111100;
40: // #1 PC = 3'b000;
41:
42: LDS = 1'b0; // iese negat
43: UDS = 1'b1; // iesenegat
44: RW = 1'b0;
45: if (IRESET == 1'b1)
46: begin
47: RESET = 0; //se inverseza
48: PC = 3'b000;
49: D[15:0] = 16'b0;
50: A[23:1] = 23'b0;
51: end
52: else
53: begin
54: if ((IPL2 == 1'b1) && (IPL1 == 1'b1)) // daca printer e BUSY
55: begin
56: if (IPL0 == 1'b1)
57: begin
58: PC = 3'b000; // program counter
59: FC2 = 1'b1; // raspuns PIACK
60: FC1 = 1'b1; // la cerere de intrerupere
61: FC0 = 1'b1; // de la MC68320
62: AS = 1'b1;

```

```

63:      A[23:22] = 2'b11;                                //la interupt ack
64:      end
65:      if (IPL0 == 1'b0)
66:      begin
67:          FC2 = 1'b1;                                    // raspuns PIACK
68:          FC1 = 1'b1;                                    // la cerere de intrerupere
69:          FC0 = 1'b1;                                    // de la MC68320
70:          AS = 1'b1;
71:          A[23:22] = 2'b11;                                //la interupt ack
72:      end
73:      end
74:  else
75:      begin
76:          if (DTACK == 1'b1)                                // pot sa transmit date
77:          begin
78:              D[7:0] = Mem[PC];
79:              D[15:8] = Mem[PC];
80:              A[5:1] = 5'b00001;                            //lpt1
81:              A[21:6] = 16'h0378;                            //adr lpt1
82:              A[23:22] = 2'b00;                                //la interupt ack
83:              PC = PC +1;
84:          end
85:      else
86:          begin
87:              D[15:0] = 16'b0;
88:              A[23:1] = 23'b0;
89:          end
90:      end
91:  end
92:  end
93:  // add your code here
94:
95: endmodule

```

```

1: module MC68230 (PAI0, PAI1, PAI2, PAI3, PAI4, PAI5, PAI6, PA7, Hdoi, H1, PC0, PC1, D, RSi, CS,
2:
3: input PA7 ;
4: input CS ;
5: input PC6 ;
6: input RW ;
7: input RESET ;
8: input CLK ;
9: input [0:7] D;
10: input [5:1] RSi ;
11: input H1 ;
12: input PC0 ;
13: input PC1 ;
14:
15: output PAI0 ;
16: output PAI1 ;
17: output PAI2 ;
18: output PAI3 ;
19: output PAI4 ;
20: output PAI5 ;
21: output PAI6 ;
22: output Hdoi ;
23: output PIC5 ;
24: output DTACKi ;
25: output ceva ;
26:
27: reg PAI0 ;
28: reg PAI1 ;
29: reg PAI2 ;
30: reg PAI3 ;
31: reg PAI4 ;
32: reg PAI5 ;
33: reg PAI6 ;
34: reg PIC5 ;
35: reg DTACKi ;
36: reg Hdoi ;
37: reg ceva ;
38:
39: // add your declarations here
40:
41: always @(posedge CLK)
42: begin
43:
44: if (RSi == 5'b00001)
45: begin
46: ceva = 1'b1 ;
47: end
48: else
49: begin
50: ceva = 1'b0 ;
51: end
52:
53: if (CS == 1'b1) // chip select din address decode
54: begin
55: if (RW == 1'b0 && RESET == 1'b0) // rw la de proc, si reset la fel
56: begin
57: if (PC1 == 1'b1 && PA7 == 1'b0) // PC = printer select de la printer PA7 = bus
58: begin
59: if (H1 == 1'b1) //daca are ack de la printer
60: begin
61: if (PC0 == 1'b1) // printer error
62: begin

```

```

63:         DTACKi = 1'b0;
64:     end
65:     else if (PC0 == 1'b0)
66:     begin
67:         if (RSi == 5'b00001)                // daca e adresa
68:         begin
69:             DTACKi = 1'b1;
70:             PIC5 = 1'b0;                    //IRQ e pe 0
71:             PAI0 = D[0];                    //trimit date
72:             PAI1 = D[1];
73:             PAI2 = D[2];
74:             PAI3 = D[3];
75:             PAI4 = D[4];
76:             PAI5 = D[5];
77:             PAI6 = D[6];
78:             Hdoi = 1'b1;                    //data srtrobe
79:         end
80:     else
81:     begin
82:         PIC5 = 1'b0;                        //trimit IRQ
83:         Hdoi = 1'b0;
84:         DTACKi = 1'b1;
85:     end
86: end
87: end
88: else if (H1 == 1'b0)
89: begin
90:     PIC5 = 1'b0;                            //trimit IRQ
91:     Hdoi = 1'b0;
92:     DTACKi = 1'b0;
93: end
94: end //pc1
95: else if (PC1 == 1'b0 && PA7 == 1'b1)
96: begin
97:     PIC5 = 1'b0;                            //trimit IRQ
98:     Hdoi = 1'b0;
99:     DTACKi = 1'b0;
100: end
101: end //rw
102: else if (RW == 1'b1 && RESET == 1'b1)        // daca nu e pe write
103: begin
104:     PIC5 = 1'b0;                            //trimit IRQ
105:     Hdoi = 1'b0;
106:     DTACKi = 1'b0;
107: end
108: end // cs
109: else if (CS == 1'b0)                        // nu e selectat controlerul
110: begin
111:     Hdoi = 1'b0;                            // zero pe date send -> print
112:     DTACKi = 1'b0;                            // nu mai cere date de la proc
113: end
114: end
115:
116: // add your code here
117:
118: endmodule

```