

```

1  TITLE "LPT_OUT modul obsługi wyjściowego łącza równoległego";
2  %
3  Modul podłączony do układu SML: 235_DB25F,
4  odbierający z szyny danych 8-bitowa liczba U2
5  i wysyłający ją w postaci ZDDD przez port LPT do drukarki.
6  (Z - znak, D - cyfra dziesiętna)
7  %
8  INCLUDE "U2_TO_ASCII.inc";
9
10 CONSTANT strobe_time = 10;  % 0,5us dla !STROBE  %
11 % 0,5u / 0,05u = 10  %
12
13 CONSTANT dev_addr    = H"10";  % adres tego urządzenia w IO  %
14 CONSTANT cr          = H"0D";  % ASCII dla CR  %
15 CONSTANT lf          = H"0A";  % ASCII dla LF  %
16
17 SUBDESIGN LPT_OUT
18 (
19     GEN          : input;  % zegar 20MHz  %
20     RESET        : input;  % sygnał resetu mikrokontr.  %
21
22     % sygnały szyny wewnętrznej mikrokontrolera  %
23     A[7..0]      : input;  % linie adresowe szyny  %
24     D[7..0]      : input;  % linie danych szyny  %
25     IORQ         : input;
26     WR           : input;
27     WAIT         : output;
28
29     % sygnał powiadamiający o gotowości drukarki, aktywny '1'  %
30     PRN_READY    : output;
31
32     % sygnały do komunikacji z łączem LPT  %
33     LCTRL[7..0]  : output;  % SV2: nSI,nI,nAF,nS do LPT  %
34     LSTAT[7..0]  : input;  % SV3: BS,PE,nF,S z LPT  %
35     LDATASYN[7..0] : output;  % SV5: dane do LPT  %
36
37     % sygnały debug do symulacji %
38     V_BPRINT     : output;
39     V_APRINT     : output;
40     V_CSTIME[5..0] : output;
41 )
42 VARIABLE
43     LDATA[7..0]  : DFF;  % synchronizowany sygnał LPT::D  %
44     BPRINT       : NODE;  % komenda z szyny: drukuj  %
45     APRINT       : NODE;  % komenda od AUT_BUS: drukuj  %
46     WAIT_I       : NODE;  % wewnętrzny sygnał WAIT  %
47     nWAIT_SYN    : DFF;  % synchronizowany sygnał !WAIT  %
48     DIN[7..0]    : NODE;  % szyna danych wejściowych  %
49     PRNT_OK      : NODE;  % czy drukarka ok?(!PE & nF & S)  %
50     BUSY         : NODE;  % sygnał BUSY od drukarki  %
51     STROBE       : NODE;  % wewnętrzny sygnał STROBE  %
52     nSTROBE_SYN  : DFF;  % synchronizowany sygnał !STROBE  %
53     TRI_DI[7..0] : TRI;  % bufor 3stanowy wyjścia na szynę D  %
54     CSTIME[5..0] : DFF;  % zlicza czas strobe  %
55     Z[7..0]      : NODE;  % kod ASCII znaku: 2D='-',2B='+'  %

```

```

56      D2[7..0]      : NODE;      % kod ASCII cyfry: 30..39      %
57      D1[7..0]      : NODE;
58      D0[7..0]      : NODE;
59      U2A            : U2_TO_ASCII; % konwerter U2->ASCII(ZDDD) %
60
61      % automat obsługujący wydruk pojedynczego znaku      %
62      AUT_PRINT      : machine of bits (QP[1..0])
63                      with states (SP0=B"00", SP1=B"01",
64                      SP2=B"10", SP3=B"11");
65
66      % automat obsługujący szynę i sterujący logiką wydruku %
67      AUT_BUS        : machine of bits (QB[3..0])
68                      with states (IDL=B"0000",
69                      CRA=B"0001",
70                      CRB=B"0010",
71                      ZA =B"0011",
72                      ZB =B"0100",
73                      D2A=B"0101",
74                      D2B=B"0110",
75                      D1A=B"0111",
76                      D1B=B"1000",
77                      D0A=B"1001",
78                      D0B=B"1010",
79                      LFA=B"1011",
80                      LFB=B"1100",
81                      WBU=B"1101");
82  BEGIN
83      % DEBUG %
84      V_BPRINT = BPRINT;
85      V_APRINT = APRINT;
86      V_CTIME[] = CTIME[];
87
88      % podłączenie przerzutników DFF i automatów      %
89      nWAIT_SYN.clk   = GEN;
90      nWAIT_SYN.clrn   = RESET;
91      nWAIT_SYN       = !WAIT_I;
92      LDATA[].clk     = GEN;
93      LDATA[].clrn     = RESET;
94      nSTROBE_SYN.clk = GEN;
95      nSTROBE_SYN.clrn= RESET;
96      CTIME[].clk     = GEN;
97      CTIME[].clrn     = RESET;
98      AUT_PRINT.clk    = GEN;
99      AUT_PRINT.reset  = !RESET;
100     AUT_BUS.clk      = GEN;
101     AUT_BUS.reset    = !RESET;
102
103     % podłączenie układu konwersji U2->ASCII(Z,D2,D1,D0) %
104     U2A.U2I[]        = DIN[];
105     Z[]              = U2A.Z[];
106     D2[]             = U2A.D2[];
107     D1[]             = U2A.D1[];
108     D0[]             = U2A.D0[];
109
110     % podłączenie linii komunikacji z łączem LPT      %

```

```

111     LDATASYN[ ]      = LDATA[ ];
112     BUSY              = LSTAT[4];
113
114     % !PError & nFault & Select                                     %
115     PRNT_OK           = !LSTAT[2] & LSTAT[1] & LSTAT[0];
116     PRN_READY         = PRNT_OK;
117
118     % zapewnienie STROBE=VCC podczas startu i RESETu               %
119     nSTROBE_SYN       = !STROBE;
120
121     % nSelectIn=0, nInit=1, nAutoFd=1                               %
122     LCTRL[ ]          = (1,1,1,1,0,1,1,!nSTROBE_SYN);
123
124     % zdekodowanie rozkazu wyslania danych do LPT                  %
125     if (A[ ]==dev_addr & IORQ==GND & WR==GND)
126         then BPRINT = VCC;
127         else BPRINT = GND; end if;
128
129     % wczytanie danych z szyny D[ ]                                %
130     TRI_DI[ ].oe      = BPRINT;
131     TRI_DI[ ].in      = D[ ];
132     DIN[ ]            = TRI_DI[ ];
133
134     % wystawienie sygnału WAIT na szynie                            %
135     WAIT              = !nWAIT_SYN;
136     %WAIT             = TRI(!nWAIT_SYN, BPRINT); % % (in, oe) %
137
138     % automat obsługujący szynę i sterujący logiką wydruku        %
139     case AUT_BUS is
140         % oczekiwanie na dane z szyny %
141         when IDL => if (BPRINT==VCC)then WAIT_I=GND;
142                     APRINT=VCC; LDATA[ ]=cr; AUT_BUS=CRA;
143                     else WAIT_I=VCC;
144                     APRINT=GND; LDATA[ ]=cr; AUT_BUS=IDL;
145                     end if;
146         % oczekiwanie na pojawienie się !STROBE dla CR
147         - oznacza to, że AUT_PRINT ruszył, nie przeoczmy
148         tego sygnału, bo trwa >1 takt zegara %
149         when CRA => if (STROBE==GND)then WAIT_I=GND;
150                     APRINT=GND; LDATA[ ]=cr; AUT_BUS=CRB;
151                     else WAIT_I=GND;
152                     APRINT=VCC; LDATA[ ]=cr; AUT_BUS=CRA;
153                     end if;
154         % oczekiwanie na koniec wydruku CR %
155         when CRB => if (AUT_PRINT==SP0)then WAIT_I=GND;
156                     APRINT=VCC; LDATA[ ]=Z[ ]; AUT_BUS=ZA;
157                     else WAIT_I=GND;
158                     APRINT=GND; LDATA[ ]=cr; AUT_BUS=CRB;
159                     end if;
160         % oczekiwanie na start wydruku Z %
161         when ZA => if (STROBE==GND)then WAIT_I=GND;
162                     APRINT=GND; LDATA[ ]=Z[ ]; AUT_BUS=ZB;
163                     else WAIT_I=GND;
164                     APRINT=VCC; LDATA[ ]=Z[ ]; AUT_BUS=ZA;
165                     end if;

```

```

166      % oczekiwanie na koniec wydruku Z                                     %
167      when ZB => if (AUT_PRINT==SP0)then WAIT_I=GND;
168              APRINT=VCC; LDATA[ ]=D2[ ];AUT_BUS=D2A;
169      else WAIT_I=GND;
170              APRINT=GND; LDATA[ ]=Z[ ]; AUT_BUS=ZB;
171      end if;
172      % oczekiwanie na start druku D2                                     %
173      when D2A => if (STROBE==GND)then WAIT_I=GND;
174              APRINT=GND; LDATA[ ]=D2[ ];AUT_BUS=D2B;
175      else WAIT_I=GND;
176              APRINT=VCC; LDATA[ ]=D2[ ];AUT_BUS=D2A;
177      end if;
178      % oczekiwanie na koniec wydruku D2                                     %
179      when D2B => if (AUT_PRINT==SP0)then WAIT_I=GND;
180              APRINT=VCC; LDATA[ ]=D1[ ];AUT_BUS=D1A;
181      else WAIT_I=GND;
182              APRINT=GND; LDATA[ ]=D2[ ];AUT_BUS=D2B;
183      end if;
184      % oczekiwanie na start druku D1                                     %
185      when D1A => if (STROBE==GND)then WAIT_I=GND;
186              APRINT=GND; LDATA[ ]=D1[ ];AUT_BUS=D1B;
187      else WAIT_I=GND;
188              APRINT=VCC; LDATA[ ]=D1[ ];AUT_BUS=D1A;
189      end if;
190      % oczekiwanie na koniec wydruku D1                                     %
191      when D1B => if (AUT_PRINT==SP0)then WAIT_I=GND;
192              APRINT=VCC; LDATA[ ]=D0[ ];AUT_BUS=D0A;
193      else WAIT_I=GND;
194              APRINT=GND; LDATA[ ]=D1[ ];AUT_BUS=D1B;
195      end if;
196      % oczekiwanie na start druku D0                                     %
197      when D0A => if (STROBE==GND)then WAIT_I=GND;
198              APRINT=GND; LDATA[ ]=D0[ ];AUT_BUS=D0B;
199      else WAIT_I=GND;
200              APRINT=VCC; LDATA[ ]=D0[ ];AUT_BUS=D0A;
201      end if;
202      % oczekiwanie na koniec wydruku D0                                     %
203      when D0B => if (AUT_PRINT==SP0)then WAIT_I=GND;
204              APRINT=VCC; LDATA[ ]=lf; AUT_BUS=LFA;
205      else WAIT_I=GND;
206              APRINT=GND; LDATA[ ]=D0[ ];AUT_BUS=D0B;
207      end if;
208      % oczekiwanie na start druku LF                                     %
209      when LFA => if (STROBE==GND)then WAIT_I=GND;
210              APRINT=GND; LDATA[ ]=lf; AUT_BUS=LFB;
211      else WAIT_I=GND;
212              APRINT=VCC; LDATA[ ]=lf; AUT_BUS=LFA;
213      end if;
214      % oczekiwanie na koniec wydruku LF                                     %
215      when LFB => if (AUT_PRINT==SP0)then WAIT_I=VCC;
216              APRINT=GND; LDATA[ ]=Z[ ]; AUT_BUS=WBU;
217      else WAIT_I=GND;
218              APRINT=GND; LDATA[ ]=lf; AUT_BUS=LFB;
219      end if;
220      % oczekiwanie na wycofanie sie z szyny pol. druku                                     %

```

```
221         when WBU => if (BPRINT==GND) then      WAIT_I=VCC;
222                     APRINT=GND; LDATA[ ]=Z[ ]; AUT_BUS=IDL;
223                     else                          WAIT_I=VCC;
224                     APRINT=GND; LDATA[ ]=Z[ ]; AUT_BUS=WBU;
225                     end if;
226     end case;
227
228     % automat obsługujący wydruk pojedynczego znaku %
229     case AUT_PRINT is
230         % oczekiwanie na polecenie druku %
231         when SP0 => if (APRINT==VCC) then
232             STROBE=VCC; AUT_PRINT=SP1;
233             else
234                 STROBE=VCC; AUT_PRINT=SP0; end if;
235         % sprawdzenie stanu linii BUSY %
236         when SP1 => if (BUSY==GND) then
237             STROBE=GND; AUT_PRINT=SP2;
238             else
239                 STROBE=VCC; AUT_PRINT=SP1; end if;
240         % wystawienie !STROBE na czas strobe_time %
241         when SP2 => if (CSTIME[ ]==strobe_time) then
242             STROBE=VCC; AUT_PRINT=SP3;
243             else
244                 STROBE=GND; AUT_PRINT=SP2;
245                 if (CSTIME[ ] < strobe_time) then
246                     CSTIME[ ]=CSTIME[ ]+1;
247                 else
248                     CSTIME[ ]=CSTIME[ ];
249                 end if; end if;
250         % zaczekanie na BUSY = L %
251         when SP3 => if (BUSY==GND) then
252             STROBE=VCC; AUT_PRINT=SP0;
253             else
254                 STROBE=VCC; AUT_PRINT=SP3; end if;
255     end case;
256
257     END;
```