

Układy PLD, część 3

Oprogramowanie - PALASM

Kontynuujemy serię artykułów na temat układów PLD. Oto część trzecia poświęcona językowi PALASM - mniej znanemu niż CUPL, ale równie użytecznemu w programowaniu układów PLD.



PALASM jest językiem mniej popularnym niż CUPL. Wynika to głównie z nieco ograniczonego obszaru zastosowań. Jest to prawdopodobnie efektem polityki firmy AMD, która, promując swoje rodziny układów PLD, rozpowszechnia jednocześnie program „kompatybilny“ tylko z układami AMD i pochodnymi.

Na rys. 1 jest przedstawiony obieg informacji pomiędzy poszczególnymi fragmentami kompilatora. Jak widać, struktura obiegu informacji w PALASM'ie jest prostsza i bardziej przejrzysta niż w CUPL'u (porównaj z rys. 1 w EP 12/93, str.27). Końcowe efekty - ilość i jakość

plików dokumentacyjnych i symulacyjnych są porównywalne.

PALASM oferuje nieco mniejsze w stosunku do CUPL'a możliwości opisu układów logicznych. Można wręcz stwierdzić, że jedyną metodą opisu są równania boolowskie - z niewielkimi modyfikacjami, gdyż rozróżnione są równania kombinacyjne, rejestrowe, zatraskowe oraz funkcyjne. Te ostatnie umożliwiają następujące operacje:

- wprowadzenie indywidualnych zegarów dla poszczególnych przerzutników (opcja - nie dla wszystkich PLD jest to możliwe);
- ustawianie (SET) i kasowanie (RESET) przerzutników (uwaga j.w.);

- sterowanie wyjściowym buforem trójstanowym.

Oprócz tego dostępne są funkcje opisujące układy automatów. Podstawą opisu może być graf z definicjami kolejnych stanów lub tablica przejść - wyjść (jest to odpowiednik tabeli prawdy dla układu sekwencyjnego). Stany są kodowane w następujący sposób:

Nazwa Stanu = Wy_1*....
.....*Wy_2*Wy_N.

Niezależnie od tego konieczne jest zdefiniowanie funkcji przejść - wyjść, co osiąga się za pomocą równań przejść:

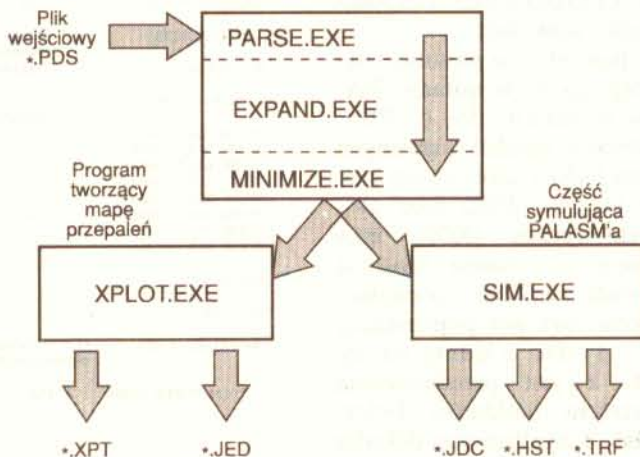
Nazwa Stanu := Warunek1 ->
Kolejny Stan + Warunek2 ->
..... +

Poprawnie zdefiniowana funkcja wymaga jeszcze zdefiniowania warunków Warunek_1...Warunek_N - definicje określają stany wejść, dla których są one spełnione.

Odbywa się to za pomocą następujących równań:

Warunek_N = We1 *
+ WeN *

Wszystkie wymienione powyżej deklaracje są podawane w pliku wejściowym *.PDS za polem tytułowym (nagłówkiem) i poprzedzone są słowami kluczowymi STATE (opis stanów i definiowanie funkcji przejść - wyjść) oraz CONDITIONS (równania warunków).



Rys. 1. Struktura obiegu informacji w kompilatorze PALASM

:PALASM Design Description

```

Declaration Segment
TITLE      Plik wejsciowy
PATTERN    A
REVISION   1.0
AUTHOR     autor
COMPANY    AVT
DATE       01/12/93

CHIP _DECODER PAL16L8
*
* Powyzej znajduje sie czesc naglowkowa
*
*
* Ponizej znajduje sie czesc deklaracji
* pinow ukkladu
*
PIN Declarations
PIN 1  CLK      COMBINATORIAL ; INPUT
PIN 2  X        COMBINATORIAL ; INPUT
PIN 3  Y        COMBINATORIAL ; INPUT
PIN 4  Z        COMBINATORIAL ; INPUT
PIN 5  GO       COMBINATORIAL ; INPUT
PIN 6  STOP     COMBINATORIAL ; INPUT
PIN 10 GND      COMBINATORIAL ; INPUT
PIN 12 A        COMBINATORIAL ; OUTPUT
PIN 13 B        COMBINATORIAL ; OUTPUT
PIN 14 C        COMBINATORIAL ; OUTPUT
PIN 15 D        COMBINATORIAL ; OUTPUT
PIN 16 E        COMBINATORIAL ; OUTPUT
PIN 17 F        COMBINATORIAL ; OUTPUT
PIN 18 G        COMBINATORIAL ; OUTPUT
PIN 19 H        COMBINATORIAL ; OUTPUT
PIN 20 VCC      COMBINATORIAL ; INPUT

*
* Ponizej rozpoczyna sie pole rownan boolowskich
*
Boolean Equation Segment
EQUATIONS
/A = /X * /Y * /Z
A.TRST = GO * /STOP

/B = /X * /Y * Z
B.TRST = GO * /STOP

/C = /X * Y * /Z
C.TRST = GO * /STOP

*
* Dla ukkladu kombinacyjnego w dalszej czesci
* pliku znajduje sie segment symulacji, dla
* ukkladu sekwencyjnego dwa dodatkowe pola:
* STATE (definicje stanow) oraz
* CONDITIONS (definicje warunkow).
*
Simulation Segment
SIMULATION
TRACE_ON X Y Z A B C D E F G H
SETF /X /Y /Z GO /STOP
CHECK /A B C D E F G H
SETF /X /Y Z
CHECK A /B C D E F G H
SETF /X Y /Z
CHECK A B /C D E F G H
SETF /X Y Z
CHECK A B C /D E F G H
SETF /GO /STOP ; DISABLE OUTPUTS
SETF /GO STOP ; DISABLE OUTPUTS
SETF GO /STOP ; DISABLE OUTPUTS
SETF GO /STOP ; ENABLE OUTPUTS
TRACE_OFF
    
```

List. 1. Format pliku wejsciowego dla PALASM'a

Format pliku wejsciowego kompilatora PALASM

Na **listingu 1** przedstawiono podstawowà strukturê pliku *.PDS. Podobnie jak w przypadku pliku wejsciowego dla jazyka CUPL, na poczàtku pliku znajduje siê nagłówek, w którym znajdują siê podstawowe informacje o pliku (tytuł projektu, wersja, data itp.). Pole to nie wymaga już chyba dokładniejszego omówienia.

W dalszej części znajduje siê pole deklaracji wyprowadzeń PINS i nastêpnie pole równań boolowskich - EQUATIONS.

W przypadku opisywania układu sekwencyjnego stosowane sà

:PALASM Design Description

```

Declaration Segment
TITLE      Podstawowe funkcory logiczne
PATTERN
REVISION   v.1.1
AUTHOR     P.Z.
COMPANY    BTC
DATE       10/27/93

CHIP _bramki PALCE16V8
*
* Realizacja szesciu roznych funkcjorow
* logicznych w ukkladzie PALCE16V8 (odp. GAL16V8)*
*
PIN Declarations
*
* WEJSCIA
*
; wejścia bramki AND:
PIN 1  A        COMBINATORIAL ; INPUT
PIN 2  B        COMBINATORIAL ; INPUT
; wejścia bramki NAND:
PIN 3  C        COMBINATORIAL ; INPUT
PIN 4  D        COMBINATORIAL ; INPUT
; wejścia bramki OR:
PIN 5  E        COMBINATORIAL ; INPUT
PIN 6  F        COMBINATORIAL ; INPUT
; wejścia bramki NOR:
PIN 7  G        COMBINATORIAL ; INPUT
PIN 8  H        COMBINATORIAL ; INPUT
; wejścia bramki EXOR:
PIN 9  I        COMBINATORIAL ; INPUT
PIN 10 GND      COMBINATORIAL ; PWR
PIN 11 J        COMBINATORIAL ; INPUT
; wejścia bramki EXNOR:
PIN 12 K        COMBINATORIAL ; INPUT
PIN 13 L        COMBINATORIAL ; INPUT
*
* WYJSCIA
*
; wyjścia bramki EXNOR:
PIN 14 Y6       COMBINATORIAL ; OUTPUT
; wyjścia bramki EXOR:
PIN 15 Y5       COMBINATORIAL ; OUTPUT
; wyjścia bramki NOR:
PIN 16 Y4       COMBINATORIAL ; OUTPUT
; wejścia bramki OR:
PIN 17 Y3       COMBINATORIAL ; OUTPUT
; wejścia bramki NAND:
PIN 18 Y2       COMBINATORIAL ; OUTPUT
; wejścia bramki AND:
PIN 19 Y1       COMBINATORIAL ; OUTPUT
PIN 20 VCC      COMBINATORIAL ; PWR

*
* Boolean Equation Segment
*
EQUATIONS
Y1 = A * B
Y2 = / (C * D)
Y3 = E + F
Y4 = / (G + H)
Y5 = I ++ J
Y6 = / (K ++ L)
    
```

List. 2a. Plik wejsciowy (*.PDS) dla PALASM'a. Realizacja bramek logicznych.

pola definicji stanów STATE oraz warunków CONDITIONS (wspomniano o tym powyżej).

Należy pamiętać o pewnej różnicy występującej w notacji PALASM'a w stosunku do CUPL'a. Inne sà symbole działañ logicznych - tzn. iloczyn (AND) zapisuje siê jako „*”, sumê logiczną (OR) jako „+”, różnicê symetrycznà (XOR) jako „+:/”, negacjê „/”. Nieco inne sà takzê pozostałe symbole „porzàdkowe”, np. komentarz jest poprzedzany znakiem „;” i tylko z jednej strony

Na **list. 2** jest przedstawiona realizacja sześciu funkcjorów logicznych, na **list. 3** zrealizowano dekoder kodu binarnego na kod wyświetlacza 7-segmentowego, a na **list. 4**

PALASM4 PAL ASSEMBLER - MARKET RELEASE 1.4 (2-5-92)
(C) - COPYRIGHT ADVANCED MICRO DEVICES INC., 1992

TITLE :Podstawowe funkcory logiczne
PATTERN
REVISION: v.1.1
AUTHOR P.Z.
COMPANY: BTC
DATE 10/27/93

```

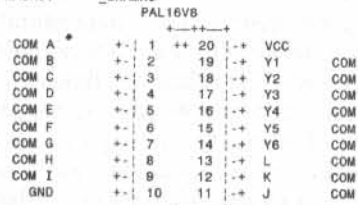
PAL16V8
_BRAMKI
0123 4567 8901 2345 6789 0123 4567 8901
0 X-X- - - - - - - -
1 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
2 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
3 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
4 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
5 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
6 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
7 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
8 X- X- - - - - - -
9 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
10 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
11 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
12 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
13 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
14 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
15 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
16 - - - - -X- -X- - - -
17 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
18 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
19 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
20 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
21 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
22 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
23 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
24 - - - - -X- -X- - - -
25 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
26 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
27 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
28 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
29 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
30 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
31 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
32 - - - - -X- -X- - - -
33 - - - - -X-X - - - -
34 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
35 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
36 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
37 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
38 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
39 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
40 - - - - -X- -X- - - -
41 - - - - -X- -X- - - -
42 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
43 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
44 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
45 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
46 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
47 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
48 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
49 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
50 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
51 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
52 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
53 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
54 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
55 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
56 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
57 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
58 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
59 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
60 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
61 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
62 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
63 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
    
```

SUMMARY

OUTPUT PINS:	11111111
	23456789
POLARITY FUSES:	-XX-
OUTPUT PINS:	11111111
	23456789
SLO FUSES:	-XXXXXX
OUTPUT PINS:	11111111
	23456789
PTERM DIS. FUSES:	-----
PTERM DIS. FUSES:	-----
PTERM DIS. FUSES:	-----
PTERM DIS. FUSES:	-----
PTERM DIS. FUSES:	-----
PTERM DIS. FUSES:	-----
PTERM DIS. FUSES:	-----
SG0 FUSE:	-
SG1 FUSE:	X
SIGNATURE FUSES:	XXXXXXXXXXXXXXXXXXXXXXXXXXXX
	XXXXXXXXXXXXXXXXXXXXXXXXXXXX
TOTAL FUSES BLOWN	= 313

Rys. 2b. Plik dokumentacyjny PALASM'a z mapà przepaleni

```
TITLE: Podstawowe funkcory logiczne
PATTERN:
REVISION: v.1.1
AUTHOR: P.Z.
COMPANY: BTC
DATE: 10/27/93
MACRO: _BRAMKI
```



List. 2c. Wyprowadzenia układu z list. 2a (bramki logiczne)

;PALASM Design Description

```

;----- Declaration Segment -----
TITLE Dekoder NKB->Hex (wyswietla 0-F).
PATTERN
REVISION 1.4
AUTHOR P.Z
COMPANY BTC
DATE 10/27/93
CHIP _HEX_DEC PALCE16V8
  
```

```

;-----
; Dekoder sterujacy wyswietlaczem
; ze wspolna anoda.
; Wyswietla znaki 0-9 oraz A-F.
;-----
  
```

;----- PIN Declarations -----

```

;-----
; Wejscia danej w kodzie dwójkowym
;-----
PIN 1 A3 COMBINATORIAL ; INPUT
PIN 2 A2 COMBINATORIAL ; INPUT
PIN 3 A1 COMBINATORIAL ; INPUT
PIN 4 A0 COMBINATORIAL ; INPUT
PIN 10 GND ; PWR
  
```

```

;-----
; Wyjscia sterujace segmentami wyswietlacza
;-----
  
```

```

PIN 13 b COMBINATORIAL ; OUTPUT
PIN 14 b COMBINATORIAL ; OUTPUT
PIN 15 c COMBINATORIAL ; OUTPUT
PIN 16 c COMBINATORIAL ; OUTPUT
PIN 17 e COMBINATORIAL ; OUTPUT
PIN 18 f COMBINATORIAL ; OUTPUT
PIN 19 g COMBINATORIAL ; OUTPUT
PIN 20 VCC ; PWR
  
```

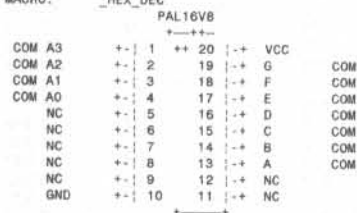
;----- Boolean Equation Segment -----

```

EQUATIONS
a = A0 * /A1 * A2 * A3
+ A0 * /A1 * /A2 * /A3
+ /A0 * /A1 * A2 * /A3
+ /A0 * /A1 * /A2 * A3
b = /A0 * /A1 * A2 * A3
+ A0 * /A1 * A2 * /A3
+ /A0 * A1 * A2
+ A0 * A1 * A3
c = A0 * A1 * A2 * A3
+ /A0 * A1 * /A2 * /A3
+ /A0 * A2 * A3
d = /A0 * A1 * /A2 * A3
+ A0 * /A1 * /A2 * /A3
+ /A0 * /A1 * A2 * /A3
+ A0 * A1 * A2
e = A0 * /A1 * /A2 * A3
+ A0 * /A2 * /A3
+ /A1 * A2 * /A3
+ A0 * A1 * A2 * /A3
f = A0 * /A1 * A2 * A3
+ A0 * /A2 * /A3
+ /A0 * A1 * /A2 * /A3
+ A0 * A1 * A2 * /A3
g = /A1 * /A2 * /A3
+ A0 * A1 * A2 * A3
  
```

List. 3a. Plik wejściowy PALSM'a. Realizacja dekodera wyświetlacza.

```
TITLE: Dekoder NKB->Hex (wyswietla 0-F).
PATTERN:
REVISION: 1.4
AUTHOR: P.Z
COMPANY: BTC
DATE: 10/27/93
MACRO: _HEX_DEC
```



List. 3b. Wyprowadzenia układu dekodera

;PALASM Design Description

```

;-----
; LICZNIK DEKADOWY Z SYNCHRONICZNYM KASOWANIEM *
; I WEJSCIEM ZMIANY KIERUNKU LICZENIA
;-----
;----- Declaration Segment -----
  
```

```

TITLE List8.PDS
PATTERN A
REVISION 1.5
AUTHOR P.Z.
COMPANY AVT
DATE 28/11/93
  
```

CHIP _LICZNIK PALCE16V8

```

;----- PIN Declarations -----
PIN 1 CLK COMBINATORIAL ; INPUT
PIN 3 RES COMBINATORIAL ; INPUT
PIN 5 DIR COMBINATORIAL ; INPUT
PIN 10 GND
PIN 14 Q3 REGISTERED ; OUTPUT
PIN 15 Q2 REGISTERED ; OUTPUT
PIN 16 Q1 REGISTERED ; OUTPUT
PIN 17 Q0 REGISTERED ; OUTPUT
PIN 19 CARRYO COMBINATORIAL ; OUTPUT
PIN 20 VCC
  
```

;----- Equations Segment -----

```

CARRYO = DIR * /RES * Q3 * /Q2 * /Q1 * Q0
+ /DIR * /RES * /Q3 * /Q2 * /Q1 * /Q0
  
```

```

; CARRYO - jest sygnałem przeniesienia przy
; przepelnieniu lub niedomiarze.
; Tworzony jest asynchronicznie - w układzie
; kombinacyjnym.
  
```

;----- State Segment -----

```

MEALY_MACHINE ; Wyznaczenie automatu Mealy'ego
START_UP := POWER_UP -> STATE0
  
```

; Poniżej znajdują się definicje stanów:

```

STATE0 = /Q3 * /Q2 * /Q1 * /Q0
STATE1 = /Q3 * /Q2 * /Q1 * Q0
STATE2 = /Q3 * /Q2 * Q1 * /Q0
STATE3 = /Q3 * /Q2 * Q1 * Q0
STATE4 = /Q3 * Q2 * /Q1 * /Q0
STATE5 = /Q3 * Q2 * /Q1 * Q0
STATE6 = /Q3 * Q2 * Q1 * /Q0
STATE7 = /Q3 * Q2 * Q1 * Q0
STATE8 = Q3 * /Q2 * /Q1 * /Q0
STATE9 = Q3 * /Q2 * /Q1 * Q0
  
```

;----- Transition Equations -----
; Poniżej znajduje się tablica przejść licznika:

```

STATE0 := GORA -> STATE1
+ DOL -> STATE9
+ RST -> STATE0
STATE1 := GORA -> STATE2
+ DOL -> STATE0
+ RST -> STATE0
STATE2 := GORA -> STATE3
+ DOL -> STATE1
+ RST -> STATE0
STATE3 := GORA -> STATE4
+ DOL -> STATE2
+ RST -> STATE0
STATE4 := GORA -> STATE5
+ DOL -> STATE3
+ RST -> STATE0
STATE5 := GORA -> STATE6
+ DOL -> STATE4
+ RST -> STATE0
STATE6 := GORA -> STATE7
+ DOL -> STATE5
+ RST -> STATE0
STATE7 := GORA -> STATE8
+ DOL -> STATE6
+ RST -> STATE0
STATE8 := GORA -> STATE9
+ DOL -> STATE7
+ RST -> STATE0
STATE9 := GORA -> STATE0
+ DOL -> STATE8
+ RST -> STATE0
  
```

;----- Output Equations -----
; Definicje stanów wyjściowych w zależności od
; aktualnego stanu licznika i stanu zewnętrznych
; sygnałów sterujących:

```

STATE0.OUTF = GORA -> /Q3 * /Q2 * /Q1 * Q0
+ DOL -> Q3 * /Q2 * /Q1 * Q0
+ RES -> /Q3 * /Q2 * /Q1 * /Q0
STATE1.OUTF = GORA -> /Q3 * /Q2 * Q1 * /Q0
+ DOL -> /Q3 * /Q2 * Q1 * /Q0
+ RES -> /Q3 * /Q2 * Q1 * /Q0
STATE2.OUTF = GORA -> /Q3 * /Q2 * Q1 * Q0
+ DOL -> /Q3 * /Q2 * Q1 * Q0
+ RES -> /Q3 * /Q2 * Q1 * /Q0
STATE3.OUTF = GORA -> /Q3 * /Q2 * Q1 * /Q0
+ DOL -> /Q3 * /Q2 * Q1 * /Q0
+ RES -> /Q3 * /Q2 * Q1 * /Q0
STATE4.OUTF = GORA -> /Q3 * Q2 * /Q1 * /Q0
+ DOL -> /Q3 * Q2 * /Q1 * /Q0
+ RES -> /Q3 * Q2 * /Q1 * /Q0
STATE5.OUTF = GORA -> /Q3 * Q2 * /Q1 * Q0
+ DOL -> /Q3 * Q2 * /Q1 * Q0
+ RES -> /Q3 * Q2 * /Q1 * /Q0
STATE6.OUTF = GORA -> /Q3 * Q2 * Q1 * /Q0
+ DOL -> /Q3 * Q2 * Q1 * /Q0
+ RES -> /Q3 * Q2 * Q1 * /Q0
  
```

```

+ RES -> /Q3 * /Q2 * /Q1 * /Q0
STATE7.OUTF = GORA -> /Q3 * /Q2 * /Q1 * /Q0
+ DOL -> /Q3 * /Q2 * /Q1 * /Q0
+ RES -> /Q3 * /Q2 * /Q1 * /Q0
STATE8.OUTF = GORA -> /Q3 * /Q2 * /Q1 * Q0
+ DOL -> /Q3 * /Q2 * /Q1 * Q0
+ RES -> /Q3 * /Q2 * /Q1 * Q0
STATE9.OUTF = GORA -> /Q3 * /Q2 * /Q1 * /Q0
+ DOL -> /Q3 * /Q2 * /Q1 * /Q0
+ RES -> /Q3 * /Q2 * /Q1 * /Q0
  
```

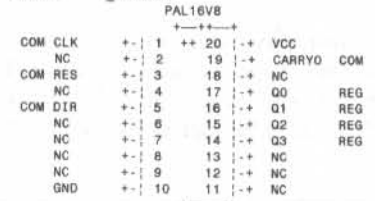
;----- Conditions -----
; Definicje warunków przejść - GORA, DOL oraz RST:

```

; D R
; I E
; R S
; ****
GORA = DIR * /RES ; 1 0 - liczenie w gore
DOL = /DIR * /RES ; 0 0 - liczenie w dol
RST = DIR * RES ; 1 1 - kasowanie
+ /DIR * RES ; 0 1 - kasowanie
  
```

List. 4a. Projekt licznika dekadowego w PALSM'ie

```
TITLE: List8.PDS
PATTERN: A
REVISION: 1.5
AUTHOR: P.Z.
COMPANY: AVT
DATE: 28/11/93
MACRO: _LICZNIK
```



Rys. 4b. Wyprowadzenia projektowanego licznika dekadowego

pokazano projekt synchronicznego licznika dekadowego. Są to ściśle odpowiedniki przykładowych listin-gów dla języka CUPL z poprzedniego artykułu (EP 12/93).

Łatwo zauważyć różnicę w sposobie opisu dekodera - PALASM zmusza do jawnego opisu funkcji wyjściowych za pomocą równań. W CUPL'u zaś ten stosunkowo żmudny i nienajłatwiejszy do weryfikacji etap można pominąć - równania i ich minimalizację wyprowadza kompilator.

Efekt pracy programów kompilujących - plik w formacie JEDEC

Każdy kompilator układów PLD tworzy plik (tekstowy lub binarny), przeznaczony dla urządzenia programującego. Zawarte są w nim wszystkie informacje niezbędne do poprawnego zaprogramowania układu. Czasami dołączane są także wektory testowe do kontroli projektu „zaszytego” w konkretny układ scalony. Żeby zapewnić wymiennność i przenośność projektów, zostały stworzone różne standardy zapisu informacji do pliku. Jednym z najbardziej popular-