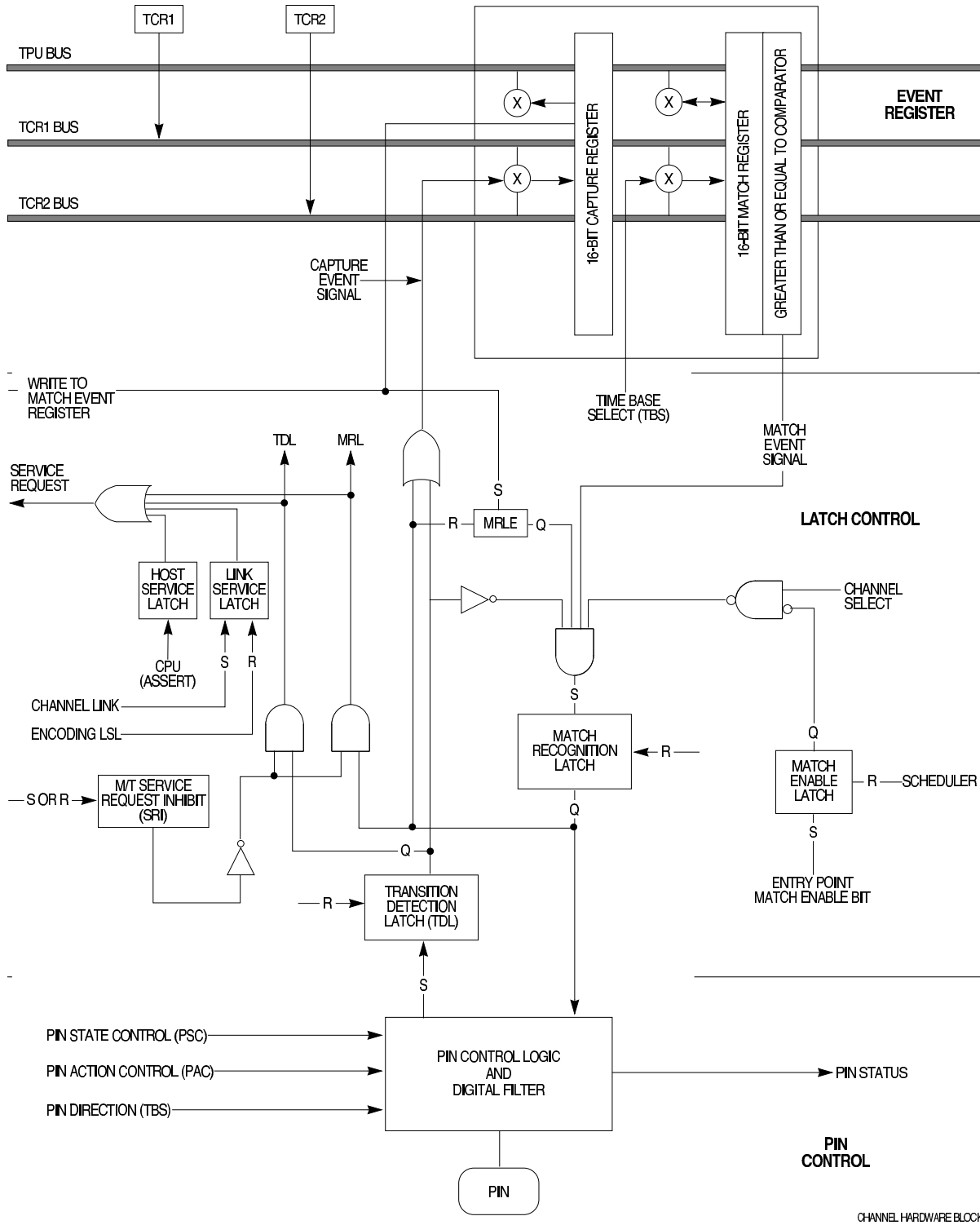


Struktura sprzętowa kanału TPU



Standardowe sterowanie kanałem TPU

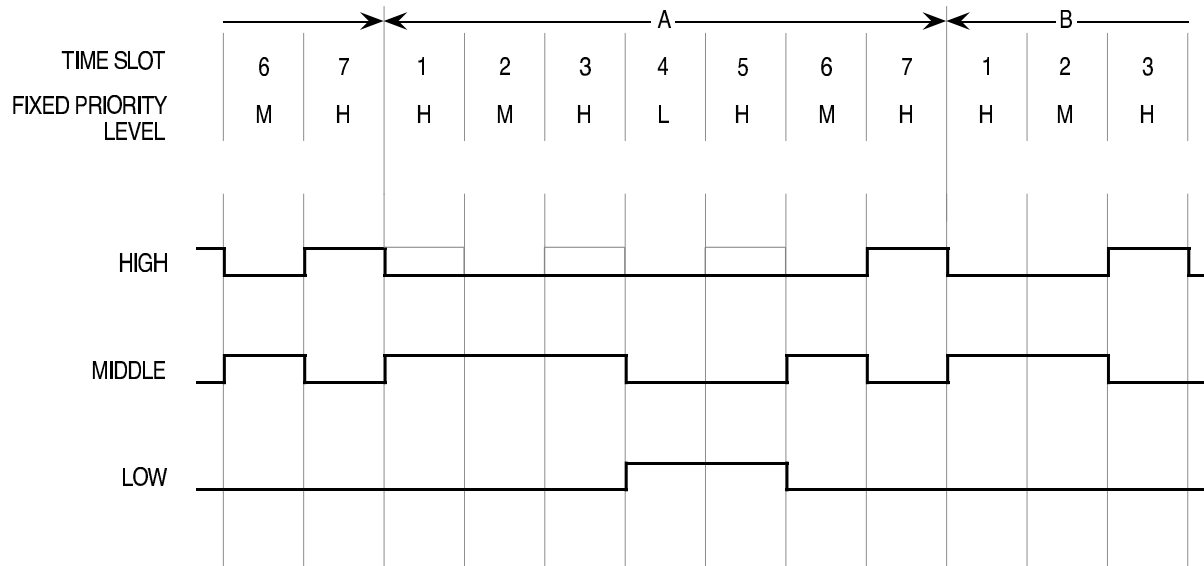
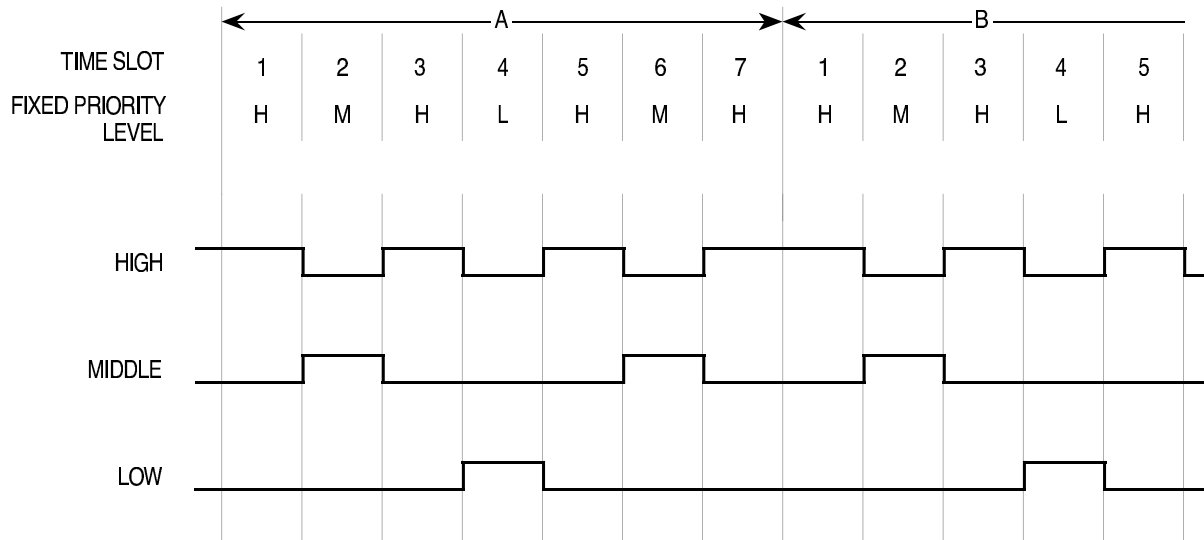
PSC - Pin State Control

PAC - Pin Action Control

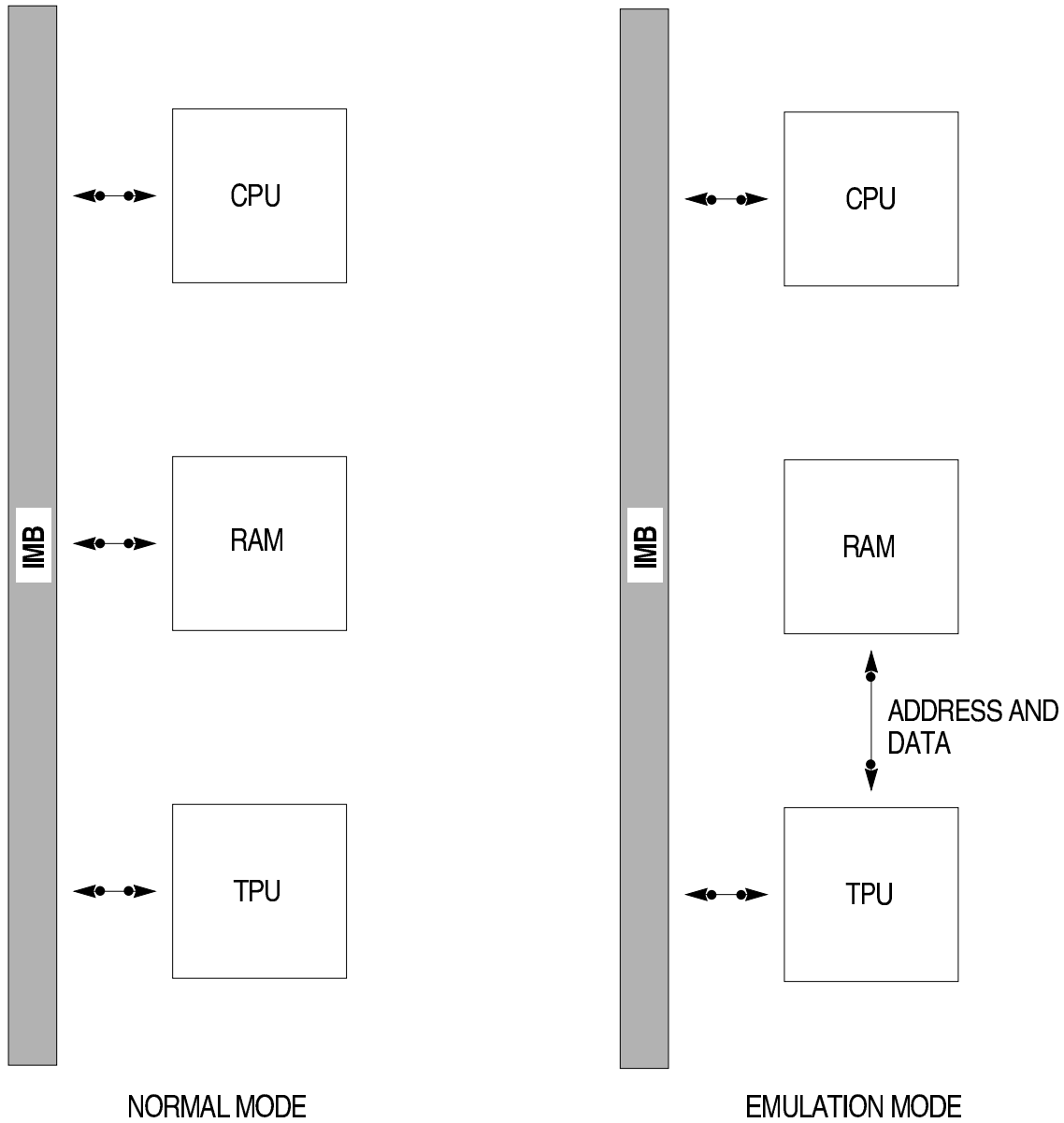
TBS - Time Base Select and Pin Direction

			Action	
TBS	PAC	PSC	Input	Output
		00		pin:=pac
		01		pin:=high
		10		pin:=low
		11		nil
	000		pac:=no_detect	pac:=no_change
	001		pac:= low_high	pac:=high
	010		pac:=high_low	pac:=low
	011		pac:=any_transition	pac:=toggle
	1xx		nil	nil
0000			tbs:=in_m1_c1	
0001			tbs:=in_m1_c2	
0010			tbs:=in_m2_c1	
0011			tbs:=in_m2_c2	
0000				tbs:=out_m1_c1
0001				tbs:=out_m1_c2
0010				tbs:=out_m2_c1
0011				tbs:=out_m2_c2
1xxx			no change	no change

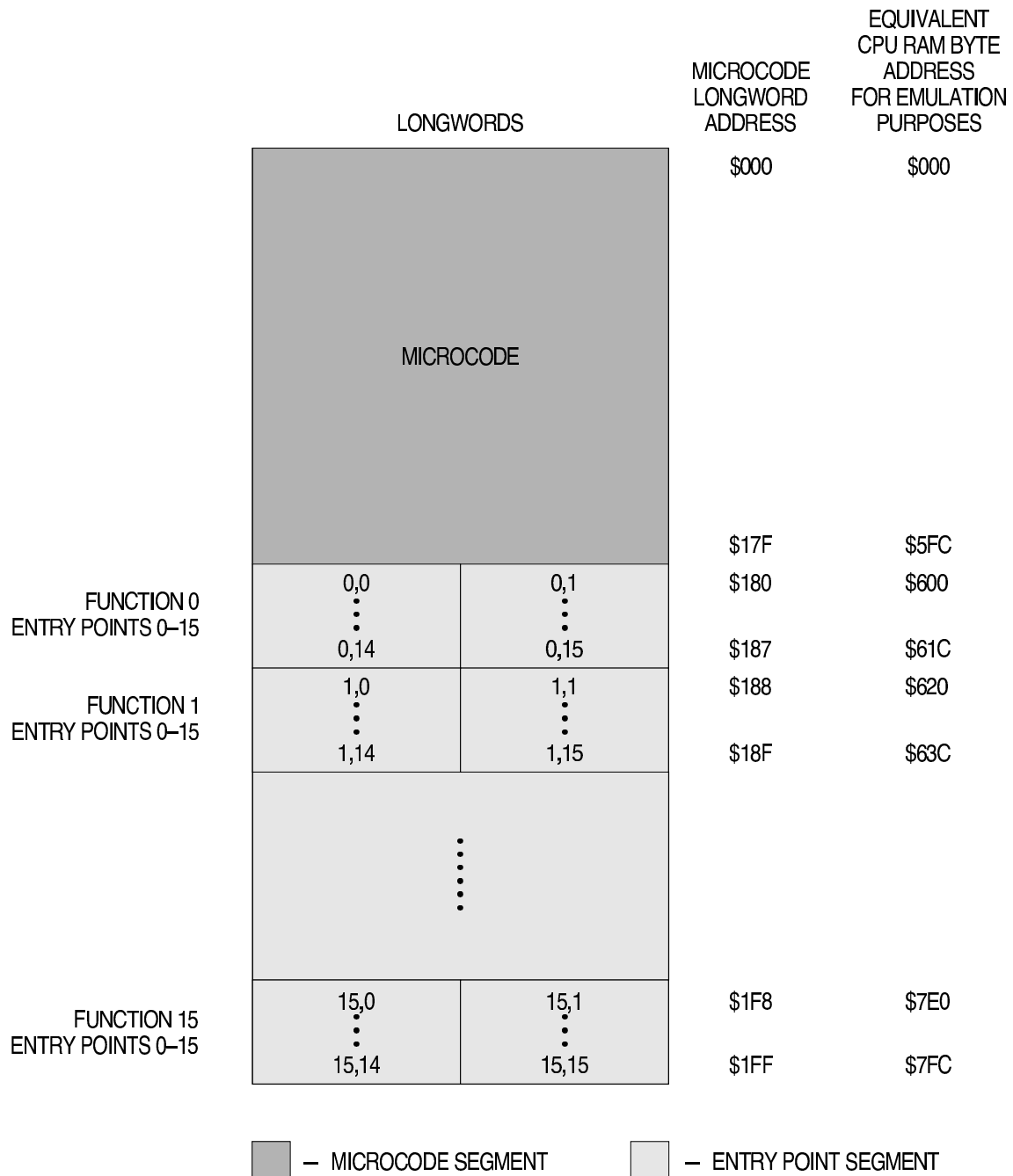
Obsługa priorytetów w TPU



Tryb pracy TPU z emulacją mikrokodu w TPURAM



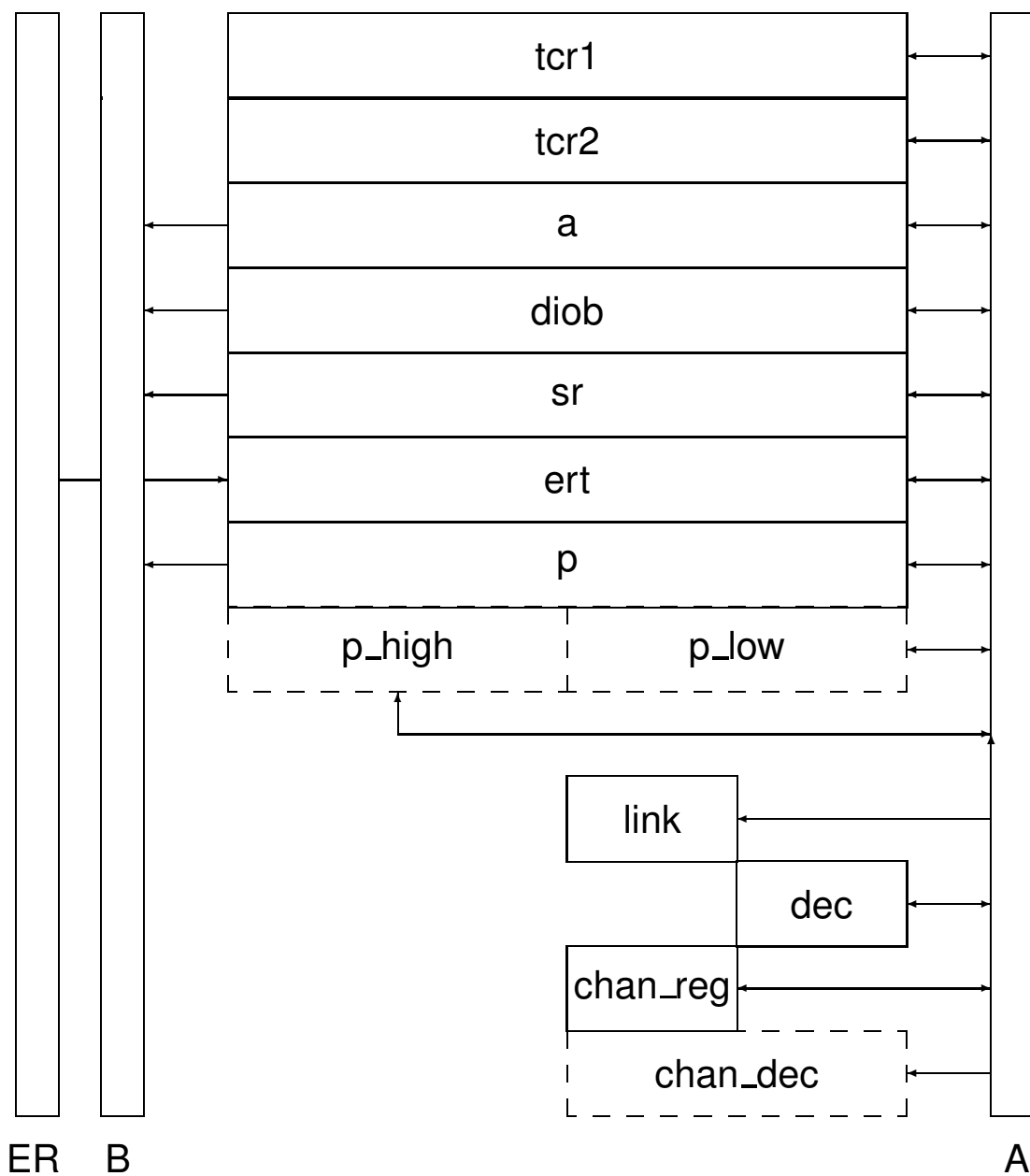
Pamięć mikro kodu TPU



Punkty startowe i stany funkcji TPU

Entry Points	Service Request Sources			Channel Conditions	
	Host Service Request (HSR)	Link Service Request (LSR)	Match/Transition Service Request (M/TSR)	Pin State	Channel Flag 0
0	01	x	x	0	x
1	01	x	x	1	x
2	10	x	x	x	x
3	11	x	x	x	x
4	00	0	1	0	0
5	00	0	1	0	1
6	00	0	1	1	0
7	00	0	1	1	1
8	00	1	0	0	0
9	00	1	0	0	1
10	00	1	0	1	0
11	00	1	0	1	1
12	00	1	1	0	0
13	00	1	1	0	1
14	00	1	1	1	0
15	00	1	1	1	1

Struktura wewnętrzna procesora TPU



Przykład mikrokodu funkcji TPU

```
(*****  
(*  
(* Function:      SQW -      RECTANGULAR WAVE  
(*  
(* Creation Date: 03/Mar/92          From: NEW  
(* Author:      Amy Dyson  
(*  
(* Description:  
(* -----  
(* SQW produces a continuous squarer wave after initialization.  
(* The user chooses the high time by writing the  
(* parameter HIGH_TIME in ram.  
(* HIGH_TIME must be between $0000-$8000.  
(*  
(* Updates:      By:      Modification:  
(* -----      ---      -----  
(* 11/Apr/93     JL      Convert to new syntax  
(* 05/Jan/99     AD      Update to Tpumasm 5.0  
(*  
(*-----*)  
(* Standard Exits Used:-      End_Of_Phase: N          End_Of_Link: Y  
(*  
(* External Files included: NONE  
(*  
(* CODE SIZE excluding standard exits = 11 LONG WORDS  
(*-----*)  
(*  
(* *****          This Revision:  REV B          *****  
(*  
(* *****          LAST MODIFIED: 11/Apr/93          BY: Jeff Loeliger *****  
(*  
(* *****
```



```

(* () () () () () () () () () () () DATA STRUCTURE () () () () () () () () () () () () () () *)
(*
(* name:                Written By:                Location  Bits:                *)
(* -----                -----                -----                *)
(* HIGH_TIME_SQW        CPU                Parameter0  0..15                *)
(*                High time of period. HIGH_TIME_SQW cannot                *)
(*                be greater than $8000.                *)
(*                *)
(* hsr1   hsr0          Action                *)
(* ----   ----          -----                *)
(*  1     1            Initialize continuous square wave                *)
(*                *)
(*                *)
(* Links Accepted: NO                Links Generated: NO                *)
(*                *)
(* Interrupts Generated After:        No interrupts generated                *)
(*                *)
(* () () () () () () () () () () () () () () () () () () () () () () () () () () () () *)

```

```
%macro HIGH_TIME_SQW    'prm0'.
```

```

%entry name = INIT_SQW; start_address *; disable_match;
cond hsr1=1, hsr0=1;
ram p <- @HIGH_TIME_SQW.
    chan                TBS := out_m1_c1,
                        PAC := toggle,
                        PIN := high,
                        enable_mtsr.
    au                  ert := tcr1 + p;
    chan                write_mer,
                        neg_mrl, neg_tdl, neg_lsl;
                        end.

```

```
%entry name = INIT_SQW; start_address *; disable_match;
cond hsr1=0, hsr0=0, m/tsr=1;
ram p <- @HIGH_TIME_SQW.
```

```
au          ert := ert + p;
chan        write_mer,
            neg_mrl, neg_tdl, neg_lsl;
end.
```

```
(*****
(* UNUSED ENTRIES - execute an end *)
*****)
```

```
%entry name = MESSAGE_UNUSED;
start_address END_OF_PHASE;
cond hsr1 = 0, hsr0 = 1.
```

```
%entry name = MESSAGE_UNUSED;
start_address END_OF_PHASE;
cond hsr1 = 1, hsr0 = 0.
```

```
%entry name = MESSAGE_UNUSED;
start_address END_OF_LINK;
cond hsr1 = 0, hsr0 = 0, m/tsr = 0, lsr = 1.
```