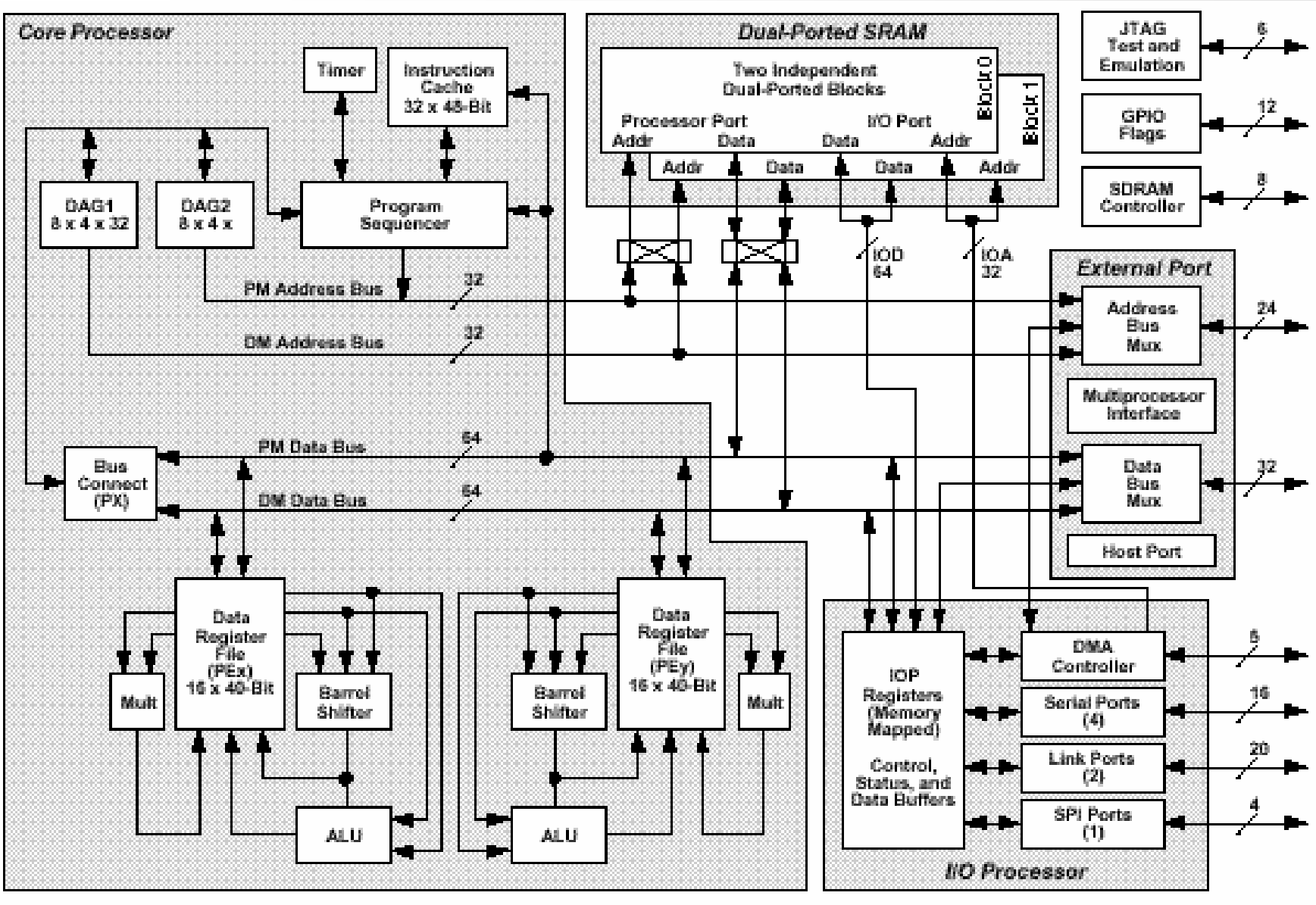
ADSP-21161 I/O Processor

Sekcja 9.

Architektura ADSP-21161



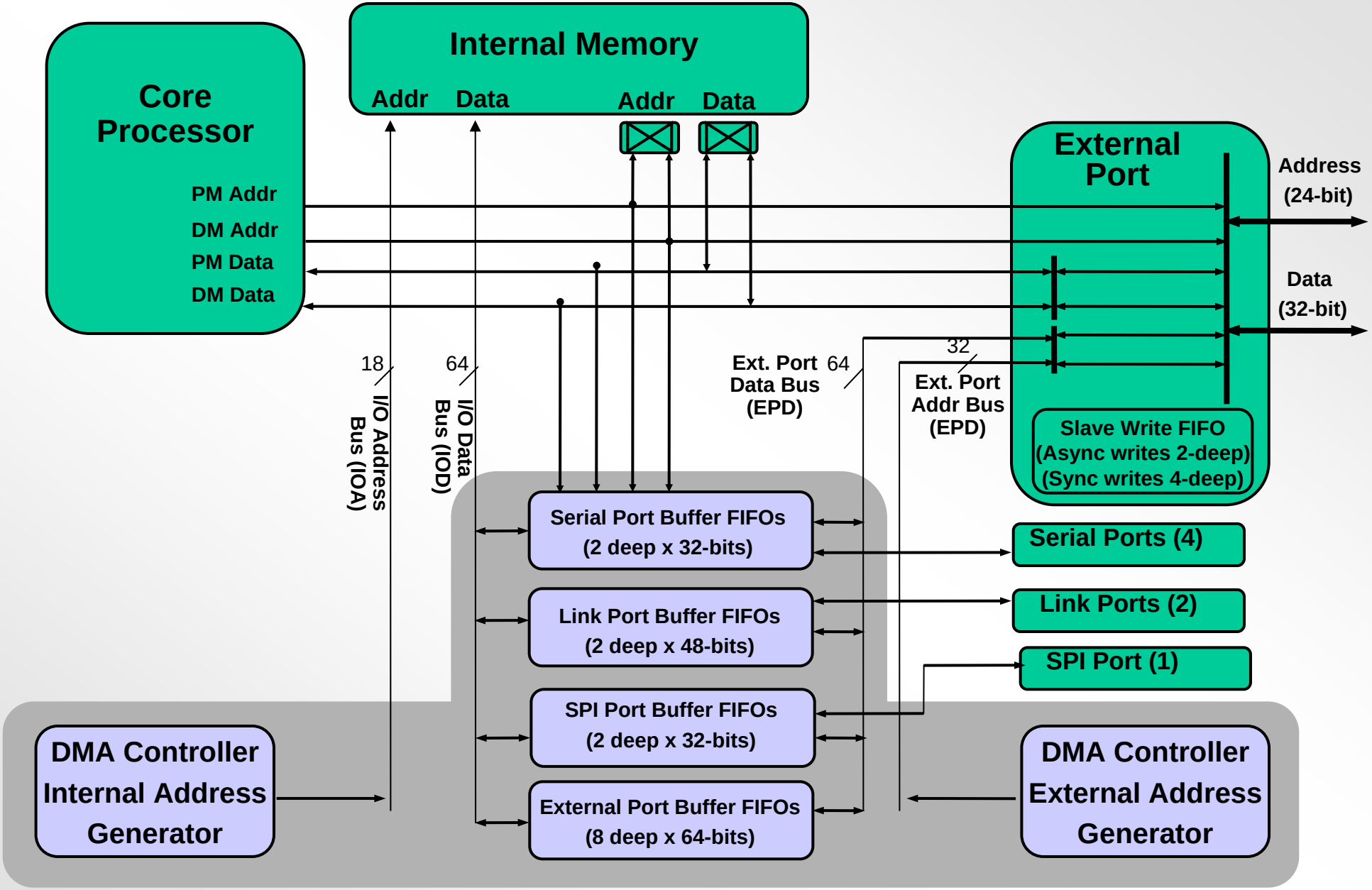
Zrownowazona architektura ADSP-21161

- **Rodzina ADSP-21161 charakteryzuje sie wysoka wydajnoscia procesora**
  - 600 Mflops w szczytowej pracy (100 MHz rdzen zegara)
  - 400 Mflops w podtrzymywanej pracy (100 MHz rdzen zegara)

- **Duza efektywnosc pamieci On-Chip**
  - 1 M-bit w ADSP-21161
  - redukuje zewnetrzny dostep i pozwala oprogramowaniu wykorzystac przewage architektury harwardzkiej

- **I/O Processor wraz z kontrolerem DMA**
  - umozliwia elastycznosc, Zero-Overhead, szybki transfer danych
  - poprzez port zewnetrzny wspomaga host/multiprocesoring/interfejs pamieci zewnetrznej
  - niezalezna maszyna przesuwania danej dla urzadzen peryferyjnych: Serial Ports, SPI i Link Ports

ADSP-21161 I/O Processor



Cechy procesora ADSP21161 I/O

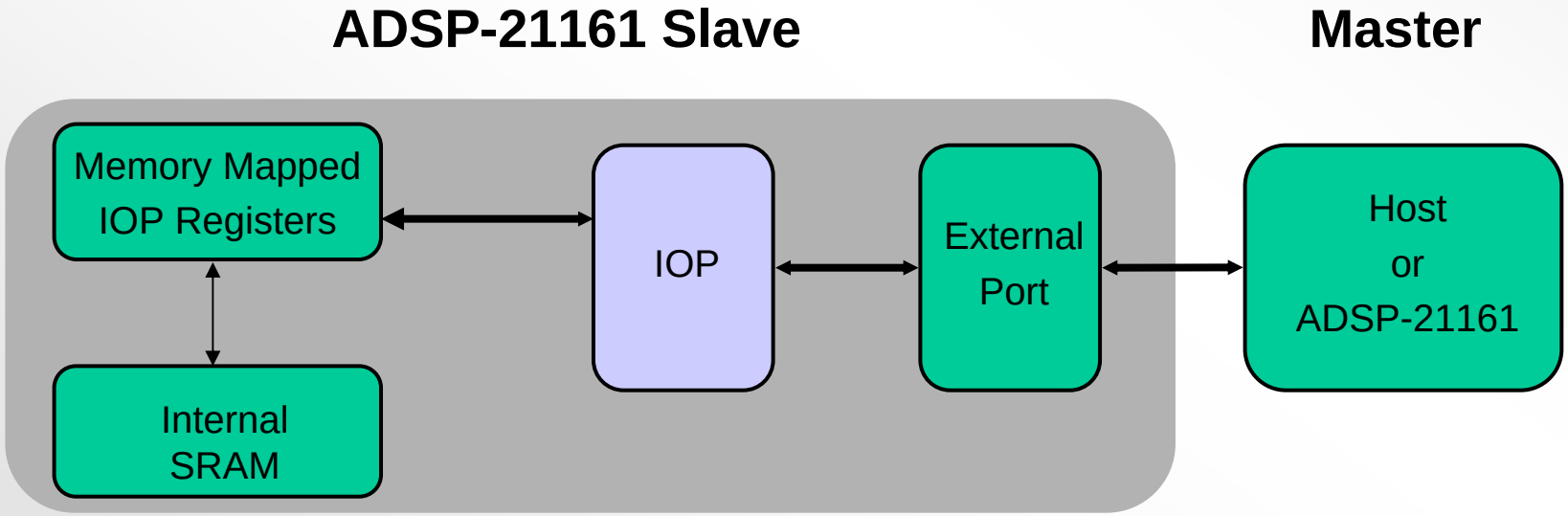
- **zintegrowany kontroler DMA do transmisji danych peryferyjnych:**
  - **port szeregowy (szeregowa dana)**
  - **Link Port (4-bity lub 8-bitow)**
  - **zewnetrzny port (8-bitow, 16-bitow, 32-bitow, 40-bity, 48-bitow, 64-bitow)**
  - **port SPI**

- **wspomaganie podstawowego transferu (widziany przez rdzen) z maksymalna przepustowoscia 600 Mbajt/s***
  - **400 MBytes/s* przez port zewnetrzny(SDRAM throughput)**
  - **200 MBytes/s* przez port zewnetrzny (non SDRAM throughput)**
  - **200 MBytes/s* przez link ports**

* 100MHz core clock

Dostep Slave

- **dostep slave do zasobow wewnetrznych ADSP-21161:**
  - **zewnetrzna szyna master moze miec bezposredni dostep do rejestrow pamieci zmapowanej w IOP**
    - **urzadzenie hostu lub master ustawia DMA na niebezposrednia transmisje z/do wewnetrznej pamieci**
  - **dostep slave jest mozliwy przez port zewnetrzny**
  - **widzialny dla rdzenia wymaga braku "interwencji"**



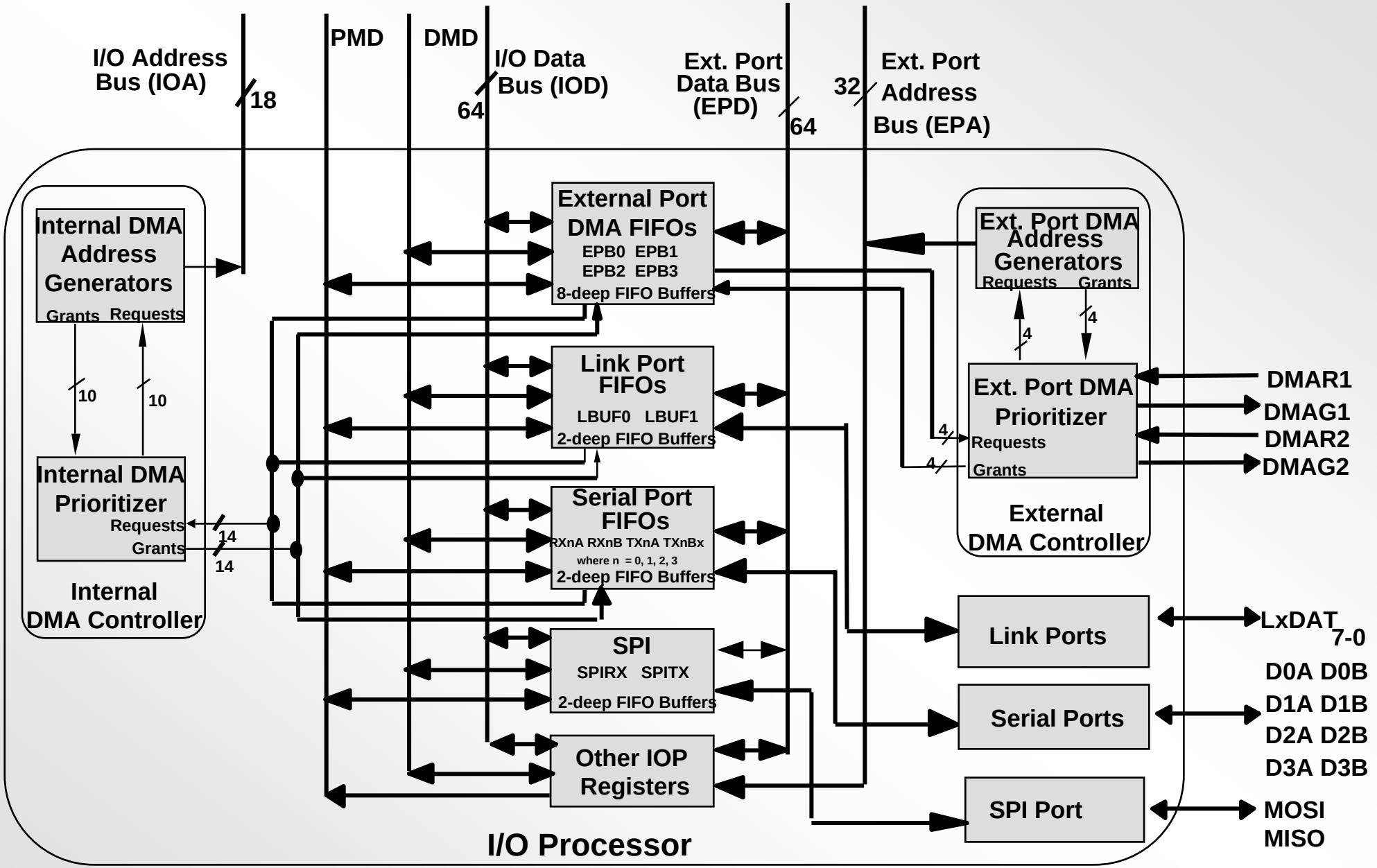
ADSP-21161 DMA

- **podstawowe przeniesienia w tempie rdzenia zegara bez interwencji rdzenia procesora**

- **sa mozliwe wszystkie nastepujace przeniesienia:**
  - **Internal Memory <-> External Memory or External Device**
  - **Internal Memory <-> Host or another SHARC**
  - **Internal Memory <-> Serial Ports**
  - **Internal Memory <-> Link Ports**
  - **Internal Memory <-> SPI Port**
  - **External Memory <-> External Device**

- **14 niezaleznych kanalow DMA**
  - **4 kanaly dla portu zewnetrznego**
  - **8 kanaly dla portu szeregowego**
  - **2 dzielone kanaly DMA pomiedzy Link Port i SPI Port**

DMA Buffers, Data Paths & Control



ADSP-21161 DMA Channel Priority

HIGHEST PRIORITY

DMA Channel 0 – Serial Port 0 A data (RX0A or TX0A)
DMA Channel 1 – Serial Port 0 B data (RX0B or TX0B)
DMA Channel 2 – Serial Port 1 A data (RX1A or TX1A)
DMA Channel 3 – Serial Port 1 B data (RX1B or TX1B)
DMA Channel 4 – Serial Port 2 A data (RX2A or TX2A)
DMA Channel 5 – Serial Port 2 B data (RX2B or TX2B)
DMA Channel 6 – Serial Port 3 A data (RX3A or TX3A)
DMA Channel 7 – Serial Port 3 B data (RX3B or TX3B)
– T*CB Chain Loading Requests**
DMA Channel 8 – Link Buffer 0/SPI Receive † (LBUF0 / SPIRX)
DMA Channel 9 – Link Buffer 1/SPI Transmit † (LBUF1) / SPITX)
DMA Channel 10 – External Port Buffer 0 † (EPB0)
DMA Channel 11 – External Port Buffer 1 † (EPB1)
DMA Channel 12 – External Port Buffer 2 † (EPB2)
DMA Channel 13 – External Port Buffer 3 † (EPB3)

LOWEST PRIORITY

**TCB Chain loading requires I/O bus and therefore requires prioritization*

†*Rotating priority can be selected for External Port Buffers and/or Link Buffers. Priority of channel block 8/9 in regards to channel block 10-13 can be rotating, as well.*

DMA Channel Programming

- **kazdy kanal DMA ma do 8 parametrow rejestrow DMA channel:**

Register	Function	Width	Description
IIx	Internal Index Register	18-bits*	Address of buffer in internal memory
IMx	Internal Modify Register	16-bits	Stride for internal buffer
Cx	Internal Count Register	16-bits	Length of internal buffer
CPx	Chain Pointer Register	19-bits*	Chain pointer for DMA chaining
GPx	General Purpose Register	18-bits	User definable
EIEPx	External Index Register	32-bits	Address of buffer in external memory
EMEPx	External Modify Register	32-bits	Stride for external buffer
ECEPx	External Count Register	32-bits	Length of external buffer

wylacznie zewnetrzny port DMA channels

*** Offset 0x40000 dla adresowania wewnetrznego**

Programowanie rejestrow IOP

- **parametry rejestrow DMA i rejestrow kontrolnych sa mapowane w rejestrach pamieci IOP, i sa dostepne wylacznie przy uzyciu DAGow:**
  - **DMAC10 jest pamiecia mapowana w adresie 0x1c**
  - **w celu zaprogramowania DMAC10, laduje sie wartosc do rejestru i uzywa DAGow w celu uzyskania dostepu do DMAC10 (i.e. “R0 = 0; DM(0x1c) = R0;”)**

- **uzycie “def21161.h” ulatwia definiowanie plikow**
  - **makra okreslaja "twardy" adres i pozycje bitu dla ADSP-21161 IOP i rejestrow systemu**
  - **przyklad: #define DMAC10 0x1c**
  - **w celu zaprogramowania DMAC10, laduje sie wartosc do rejestru i uzywa DAGow w celu uzyskania dostepu do DMAC10 (np. “R0 = 0; DM(DMAC10) = R0;”)**

- **preprocesoring asemblerowy**
  - **zastepuje symbole wartosciami z naglowka pliku**
  - **kod jest latwiejszy do odczytania, zapisania i zrozumienia**

Konwencje nazywania DMA Channel Parameter Register (DEF21161.H)

- **Serial Port Parameter Registers**
  - **IIxn, IMny, Cxn, CPxn, GPxn**
    - **Where ‘x’ = 0, 1, 2, 3 and ‘n’ = A or B**

- **Link Port / SPI Port (share DMA channel registers)**
  - **IILBx, IMLBx, CLBx, CPLBx, GPLBx (Link Port naming conventions)**
    - **Where ‘x’ = 0 or 1**
  - **IISx, IMSx, CSx, GPSx (SPI naming conventions)**
    - **Where ‘x’ = TX or RX**

- **External Port Buffer Parameter Registers**
  - **IIEPx, IMEPx, CEPx, CPEPx, GPEPx, EIEPx, EMEPx, ECEPx**
    - **Where ‘x’ = 0, 1,2, 3**

UWAGA: starsza konwencja nomenklaturowa jest wciaz dostepna w pliku DEF21161.H

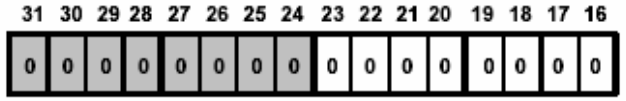
Konfigurowanie DMA

W celu ustawienia kanalu DMA dla przeniesienia DMA:

- **Wzapisac do rejestrow IIx, IMx, Cx wyszczegolnionej pamieci wewn. informacji i numeru przeniesienia**
  - **zewnetrzny port przeniesienia DMA wymaga zapisu rejestrow EIEPx, EMEPx, ECEPx**

- **aktywowanie DMA poprzez zapisanie do rejestru Channel Control (poczatek DMA):**
  - **DMACx** **External Port DMA**
  - **SPCTLx** **Serial Port DMA**
  - **LCTL** **Link Port DMA**
  - **SPICTL** **SPI Port DMA**

DMAC10 0x1c
DMAC11 0x1d
DMAC12 0x1e
DMAC13 0x1f

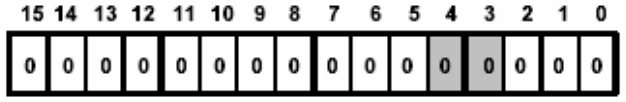


PS
Ext Port EPBx FIFO Buffer Packing Status (read-only)
000=packing complete
001=1st stage pack/unpack
010=2nd stage pack/unpack
011=3rd stage
100 = 5th stage of 8 to 48 -bit packing
101=110=111= *reserved*

MAXBL
Maximum Burst Length Select
00=burst disabled
01=burst limit of 4
10=11= *reserved*

FS
Ext. Port FIFO Buffer Status (read-only)
00=buffer empty
01=buffer-not-full
10=buffer-not-empty
11=buffer full

INT32
Internal Memory 32-bit Transfers Select
1=32-bit transfers/EPBx access width
0=64-bit transfers/EPBx access width



PRIO
External Port Bus Priority Access
1=DSP asserts PA~ for external bus access
0=PA~ not asserted

FLSH
Flush EPBx FIFO Buffers & Status
1=flush EPBx

EXTERN
External Handshake Mode Enable
1=enable, external devices to external memory
0=disable

INTIO
Single Word Interrupts for EPBx FIFO Buffers
1=enable single-wd non-DMA interrupt-driven xfers
0=disabled, FIFO fully enabled

HSHAKE
EPBx DMA Handshake Mode Enable
1=enable, 0=disable

MASTER
EPBx DMA Master Mode Enable
1=enable, 0=disable

MSWF
Most Significant Word First During Packing
1=enable, MSW first
0=disable, LSW first

DEN
Ext. Port DMA Enable
1=enable, 0=disable

CHEN
Ext. Port DMA Chaining Enable
1=enable, 0=disable

TRAN
Ext. Port EPBx Transmit/Rcv. Select
1=transmit data from intern memory
0=receive data from ext memory

DTYPE
EPBx FIFO Buffer Data Type Select
1=40/48-bit, 3-column data
0=32/64-bit, 4-column data

PMODE
Ext Port EPBx FIFO Packing Mode
000, 111= *reserved*
001=16 ext-to-32/64 int
010=16 ext-to-48 int
011=32 ext-to-48 int
100=no pack (32 ext-to-32/64 int)
101=8 ext-to-48 int
110=8 ext-to-32/64 int

Przyklad zewnetrznego portu DMA

W tym przykladzie wykonamy przeniesienie w trybie master DMA. Dana z pamieci wewnetrznej bedzie przeniesiona do pamieci zewnetrznej. Stworzylismy dwa bufory danych, jeden nazywany "source" i drugi "dest", kazdy dlugosci 10. Jeden umieszczony w wewnetrznej pamieci, drugi w zewnetrznej. Do transmisji uzyjemy DMA channel 10 (zewnetrzny port buforu 0). Wszystkie rejestry DMA channel 10 musza zainicjalizowane na transmisje. DMAC10 jest wowczas zapisywany informacja kontrolna DMA. Zapis do DMAC10 dla nie lancuchowego DMA rozpocznie przeniesienie DMA.

```
/*DMA.ASM        ADSP-21161 DMA Transfer Example */

#include      “def21161.h”

.SECTION/DM  int_mem
      .VAR  source[10];                    /* internal data buffer      */


.SECTION/DM  ext_mem
      .VAR  dest[10];                      /* external data buffer      */


.SECTION/PM                seg_pmco;
/*_____________________start of main routine_____________________________*/
start:
      r0=0;                dm(DMAC10)=r0;        /* Clear DMA Control Register        */
      r0=source;           dm(IIEP0)=r0;         /* Set internal pointer to source    */
      r0=1;                dm(IMEP0)=r0;         /* Set stride to 1 for internal buffer */
      r0=length(source);   dm(CEP0)=r0;          /* Set count to length of source     */
      r0=dest;             dm(EIEP0)=r0;         /* Set external pointer to dest      */
      r0=1;                dm(EMEP0)=r0;         /* Set stride to 1 for external buffer */
      r0=length(dest);     dm(ECEP0)=r0;         /* Set count to length of dest       */
      r0= MASTER | PMODE4 | TRAN | DEN;          /* DMAC10 = 0x0505                   */
      dm(DMAC10)=r0;                             /* dma enable, int>ext, master mode  */
                                                 /* starts the DMA                    */
wait:          idle;
               jump wait;
```

Chained DMA

- **lancuchowanie DMA jest procesem auto-linkage przeniesien DMA**
  - **ukonczenie jednego DMA ustawia inne DMA**
  - **nie wymaga sie "interwencji" rdzenia**

- **Transfer Control Block (TCB) jest wartoscia parametru danej bufora DMA przechowywanej w pamieci wewnetrznej**
  - **dla zawarcia wszystkich parametrow DMA ma zawsze dlugosc 8**
  - **wartosci moga byc inicjalizowane automatycznie lub recznie (programowane przy uruchomieniu)**

- **rejestry DMA Channel Parameter sa automatycznie ladowane przez z TCB przez IOP, gdy lancuchowanie jest aktywne**
  - **pierwsze DMA w lancuchu zaczyna sie od zapisania adresu TBC do rejestru CPx DMA channel**
  - **zaladowanie lancucha wymaga 8 cykli IOP**

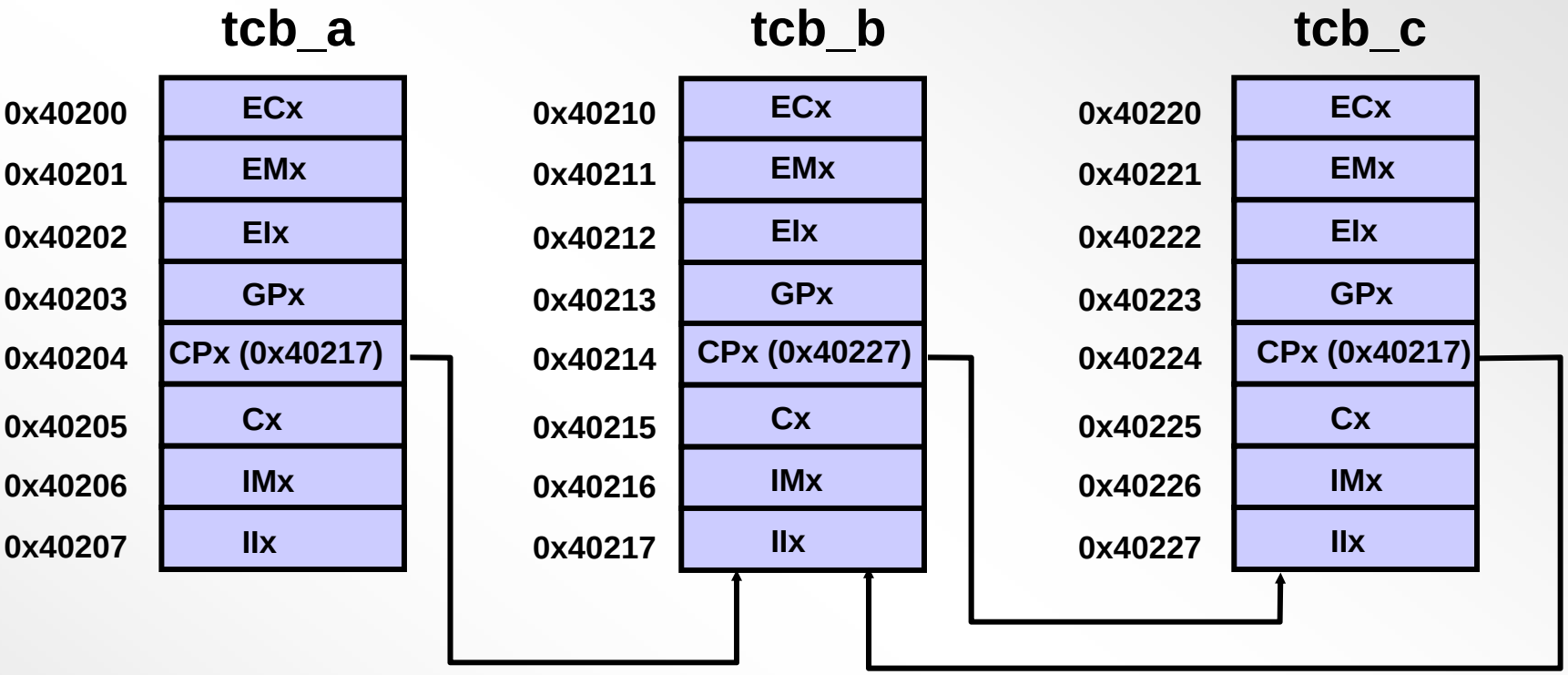
Konfigurowanie Chained DMA

W celu ustawienia kanalu DMA dla przeniesienia Chained DMA:

- **ustawic Transfer Control Blocks (TCBs) w pamieci wewn.**
- **ustawic DMA and CHAINING poprzez wpisanie rejestru Channel Control:**
  - **DMACx External Port DMA**
  - **SPCTLx Serial Port DMA**
  - **LCTL Link Port DMA**

  **** Chaining is not supported for SPI DMA**
- **zaprogramowac Chain Pointer register (CPx) adresem pierwszego TCB (poczatek DMA)**

TCBs i Chained DMA



- Transfer Control Blocks sa lokowane w pamieci wewnetrznej, ktora przechowuje we wlasciwej kolejnosci rejestr DMA. Rejestr Chain Point (CPx) przechowuje adres kolejnej TCB**.**
- wpisanie 0x40207 do CPx rozpoczyna lancuch DMA
- koniec lancucha uzyskuje sie ladujac adres CPx (bity 17:0) na przestrzeni TCB zerami

```
/*CH_DMA.ASM        ADSP-21161 Chained DMA Transfer Example*/

.SECTION/DM   int_mem;
      .VAR  source1[10];
      .VAR  tcb_a= 10, 1, dest1, 0, tcb_b + 7, 10, 1, source1;
      .VAR  source2[10];
      .VAR  tcb_b= 10, 1, dest2, 0, 0, 10, 1, source2;


.SECTION/DM   ext_mem;
      .VAR  dest1[10];
      .VAR  dest2[10];

.SECTION/PM     seg_pmco;
/*_____________________start of main routine_____________________________*/
start:
      r0=0;
      dm(DMAC10)=r0;      /* Clear DMA Control Register          */

      r0= MASTER | PMODE4 | TRAN | CHEN | DEN;          /* DMAC10 = 0x0507       */
      dm(DMAC10)=r0;      /* DMA enable, Chain enable, int->ext*/
                          /* master mode transfer                */
      r0=tcb_a+7;
      dm(CPEP0) =r0;      /* Load CP register with address of    */
                          /* 1st TCB to kick off the DMA chain   */
wait:           idle;
                jump wait;
```

Przerwania DMA

- kazdy kanal DMA ma swoje wlasne przerwanie
  - przerwania DMA sa zatrzaskiwane w rejestrze IRPTL i odmaskowane w rejestrze IMASK
  - przerwania Link Port i SPI DMA sumowane (funkcja OR) poprzez przerwanie LPISUM w rejestrze IRPTL. Sa osobno zatrzaskiwane i rozmaskowane zatrzaskiwane w rejestrze LIRPTL.
  - mozliwe jest przerwanie Global (IRPTEN w MODE1)

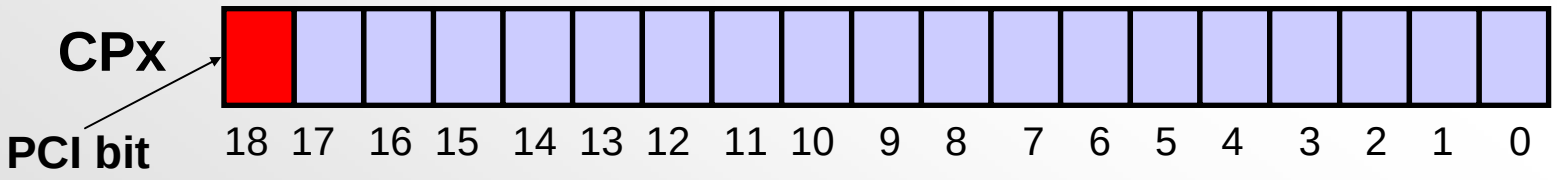
- do przerwan DMA dochodzi, gdy rejestr Count zadekrementuje do 0 (DMA complete)
  - zadanie No DMA jest generowane, gdy nastepuje wpis zera do rejestru count
  - wpis zera nie zatrzymuje DMA, ale powoduje jej dekrementacje z najwyzszej lmozliwej wartosci

- dla przerwan zewnetrznego kanalu portu DMA zarowno wewnetrzny jak i zewnetrzny licznik musza byc zdekrementowane do 0
  - gdy "packing" jest aktywne uzywa sie roznych slow liczacych

Przerwania DMA z lancuchowaniem

- **uzytkownik wybiera czy zakonczenie kazdego bloku wymaga przerwania.**
- **jest kontrolowane przez Program Controlled Interrupt (PCI) rejestru CPx (bit 18 z CPx)**
  - **PCI = 0: brak przerwania DMA jest wymagany dla obecnego zakonczenia bloku**
    - **P = 1: zadanie przerwania DMA, gdy rejestr osiaga zero**
    - **musi byc wykonane w TCB**

```
#define  PCI  0x40000     // Bit 18 set
#define  TCBMask  0x3FFFF              // Keep Lower 18 Bits
.VAR  tcb_a  =  10,  1,  dest1,  0, (tcb_b + 7) & TCBMask ,        10,  1,  source1;
      // No interrupt requested
.VAR  tcb_b  =  10,  1,  dest2,  0, (tcb_a + 7) & TCBMask | PCI,  10,  1,  source2;
      // Interrupt occurs when count reaches zero
```



Cechy zewnetrznych portow DMA

- 8 polozen 64-bitowego bufora fifo dla efektywnej transmisji danych
  - Memory mapped IOP registers (EPBx)
- 5 trybow wykonywania operacji konfigurowanych poprzez rej. DMACx
  - Slave, Master, Paced Master, Handshake, & External Handshake
- pakowanie/rozpakowywanie danych wybierane przez kanal
  - Ext:Int (8:32/64, 8:48, 16:32/64, 16:48, 32:48, 32:32/64 – “no-packing”)
- selektywny priorytet wyboru popmiedzy 4 kanalami EP
  - schemat priorytetow moze byc ustawiony lub rotacyjny (w DMACx)

Tryby zewnetrznych Portow DMA

Mode	Operation
Slave	**Internal Memory <--> EPBx**
Master	**Internal Memory <--> EPBx <--> External Memory** *uses strobes & address, no /DMAR & /DMAG*
Pace Master	**Internal Memory <--> EPBx <--> External Memory** *uses /DMAR & strobes & address, no /DMAG*
Handshake	**Int. Memory <--> EPBx <--> External Latch/Buffer** *uses /DMAR & /DMAG, no address or strobes*
External Handshake	**External Latch/Buffer <--> External Memory** *uses /DMAR & /DMAG & strobes & address*

64bitowa DMA wobec 32bitowego portu zewn.

- **kontroler DMA bedzie automatycznie probowal wykorzystac 64-bit. wewnetrzna transmisje DMA do buforow EPBx jesli zachodza nastepujace warunki:**
- **parametry EPBx DMA:**
  - **IIEPx 64-bit ustawiony (parzysty normalny adres slowa)**
  - **IMEPx = 1 lub -1**
  - **CEPx = parzyste slowo 32-bitowe**
- **rejestr kontrolny DMACx:**
  - **PMODE = 110 (8-32/64), 001 (16-32/64), 100 (32-32/64)**
  - **glebokosc bufora EPBx > 1 (mozliwa transmisja wiecej niz jednego slowa 32-bitowe)**
  - **DTYPE = 0 (typ danej nie jest instrukcja)**
  - **INT32 = 0 (nie wymusza 32-bitowej transmisji wewnetrznej)**