

Technická zpráva



Akademie věd České republiky
Ústav teorie informace a automatizace AV ČR, v.v.i.

DMA jednotka pro BCE v systémech s AXI sběrnici

|Zdeněk Pohl
zdenek.pohl@utia.cas.cz

Obsah

1. Úvod.....	2
2. Obecný popis k aplikaci.....	2
2.1 Software.....	2
2.2 Registry axi_pb_dma (axi4lite)	3
2.3 pb_frontend interface.....	4
3. Použité/Potřebné vybavení a nástroje	5
4. Jak spustit	5
5. Známé problémy a jejich odstranění.....	5
6. Obsah a popis přiloženého balíku	5

Revize

Revize	Datum	Autor	Popis změn v dokumentu
0	19.11.2012	Z.P.	Vytvoření dokumentu
1	5.12.2012	Z.P.	Revize obsahu, popis testů
2			

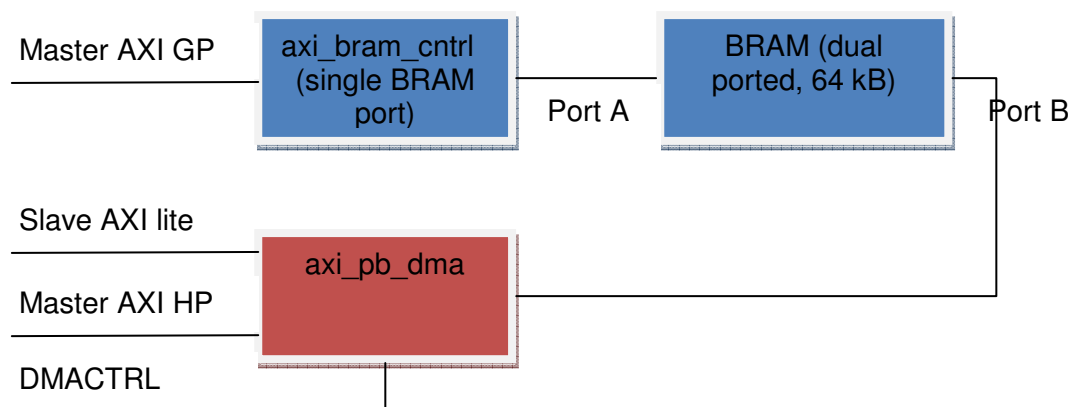
1. Úvod

Popisuje PCORE axi_pb_dma, sloužící pro datovou komunikaci nahrazující np_i_dma v systémech s AXI sběrnici.

2. Obecný popis k aplikaci

PCORE axi_pb_dma využívá pb_frontend interface k zadávání požadavků na komunikaci mezi AXI sběrnici a BRAM pamětí (max 64kB). Díky tomu je plně zpětně kompatibilní s np_i_dma. Narozdíl od np_i_dma však nabízí pouze jeden port a tudíž je potřeba jedna instance pro každý BCE. Je spoléháno na to, že paralelní přístup z více BCE jader je řešen na úrovni AXI/AMBA sběrnice. Jednotka pracuje jako master a využívá 32-bitový AXI interface.

K ověření funkce a rychlosti komunikace byl vytvořen testovací design, jehož struktura je na obrázku 1.



Obrázek 1

Jedná se o zapojení jednoho bloku BRAM o velikosti 64kB jedním portem na BRAM kontrolér pro přístup přes CPU a druhým portem na blok axi_pb_dma. Tomu je možné zadávat operace prostřednictvím registrů přístupných v paměťovém prostoru přez AXI lite interface nebo pomocí DMACTRL rozhraní přímo propojeného s PicoBlaze (v BCE).

2.1 Software

SDK workspace obsahuje 3 aplikace:

1. Aplikaci pro měření rychlosti komunikace mezi DDR, OCM (on chip memory) a BRAM pomocí ARM CPU a vestavěné DMA jednotky (test_speed). Příklad výstupu:

```
--- Entering main() ---  
Testing CPU transfer speeds
```

```

CPU cache ENABLED (speed in MB/s)
DDR->DDR, OCM, BRAM, OCM->DDR, OCM, BRAM, BRAM->DDR, OCM, BRAM

-----
|   DDR   |   OCM   |   BRAM  |
-----
DDR      |   116.4 |   116.6 |   22.2
OCM      |   118.4 |   118.4 |   22.3
BRAM     |   21.1  |   21.1  |   12.1

DMA PS330 (speed in MB/s)
DDR->DDR, OCM, BRAM, OCM->DDR, OCM, BRAM, BRAM->DDR, OCM, BRAM

-----
|   DDR   |   OCM   |   BRAM  |
-----
DDR      |   121.2 |   130.4 |  108.5
OCM      |   129.0 |   116.2 |  115.7
BRAM     |   109.8 |   113.3 |   12.1

AXI_PB_DMA speed (read):
-----
DDR offset 0: 269.0 MB/sec

AXI_PB_DMA speed (write):
-----
DDR offset 0: 308.1 MB/sec

```

Rychlost axi_pb_dma je pro 100MHz design a 4kB blok dat a je měřená na BRAM interface. Maximalní teoretická rychlost je tedy 400MB/s.

2. Aplikaci pro testování přenosu dat pomocí axi_pb_dma (test_axi_pb_dma, 8000 transakcí) Příklad výstupu na terminál:

```

OK,transaction 37100, src_DDR = 0x00115184, bram = 0x00003238, len = 0xb5c0, dst_DDR =
0x00164018, chan 4, so far 0 errors

```

Je možné vidět pořadové číslo transakce, zdrojovou DDR adresu, cílovou adresu v BRAM, cílovou adresu v DDR a použitý kanál (viz dokumentace k pb_frontend)

3. Aplikaci pro test 2 paralelních operací (test_paralel_channels). Příklad výstupu:

```

1 tests started
Waiting for 84 iterations done, max_state 3
Waiting for 76 iterations done
OK,transaction 0, src_DDR = 0x00114028, bram = 0x00000000, len = 0x8000, dst_DDR =
0x00124030, chan 0, so far 0 errors
Tests done, statistics:
  R/W errors   : 0 of 2 operations

Exiting...

```

Hodnota max_state indikuje paralelně běžící kanály 0 a 1 (bit 0 a 1 pb_frontend state register)

2.2 Registry axi_pb_dma (axi4lite)

Registry jsou identické s automaticky generovaným axi4 master PCORE, jsou však navíc rozšířeny o programování pb_frontend rozhraní (tučně):

```

Control Register (C_BASEADDR + OFFSET + 0x0):

```

```

bit 0    - Rd      (Read Request Control)
bit 1    - Wr      (Write Request Control)
bit 2    - BL      (Bus Lock Control)
bit 3    - Brst   (Burst Assertion Control)
bit 4-7  - Spare  (Spare Control Bits)
bit 4   - en signal to pb_frontend
bit 5   - we signal to pb_frontend
bit 6   - for setting up data transactions use:
           '0' - pb_frontend
           '1' - original axi4lite registers
bit 7   - determines how the pb_frontend will be programmed:
           '0' - DMACTRL
           '1' - extended axi4lite registers
Status Register (C_BASEADDR + OFFSET + 0x1):
bit 0    - Done   (Transfer Done Status)
bit 1    - Busy   (User Logic Master is Busy)
bit 2    - Error  (User Logic Master request got error response)
bit 3    - Tmout  (User Logic Master request is timeout)
bit 2-7  - Spare  (Spare Status Bits)
Address Register (C_BASEADDR + OFFSET + 0x4):
bit 0-31 - Target Address (This 32-bit value is used to populate the
IP2Bus_Mst_Addr(0:31) address bus during a Read or Write
user logic master operation)
Byte Enable Register (C_BASEADDR + OFFSET + 0x8):
bit 0-15 - Master BE (This 16-bit value is used to populate the
IP2Bus_Mst_BE byte enable bus during a Read or Write user
logic master operation for single data beat transfer)
pb_frontend addr (C_BASEADDR + OFFSET + 0xA):
  bit 0-7 - Address value to pb_frontend
pb_frontend dout (C_BASEADDR + OFFSET + 0xB):
  bit 0-7 - Dout value to pb_frontend (input data)
Length Register (C_BASEADDR + OFFSET + 0xC):
bit 0-15 - Transfer Length (This 16-bit value is used to populate the
IP2Bus_Mst_Length(0:15) transfer length bus which specifies
the number of bytes (1 to 65532) to transfer during user
logic master Read or Write fixed length burst operations)
pb_frontend din (C_BASEADDR + OFFSET + 0xE):
  bit 0-7 - Din from pb_frontend (output data, read only)
Go Register (C_BASEADDR + OFFSET + 0xF):
bit 0-7 - Go Port (Write to this byte address initiates the user
logic master transfer, data key value of 0x0A must be used)

Note: OFFSET may be different depending on your address space
configuration, by default it's either 0x0 or 0x100. Refer to IPIF
address range array for actual value.

```

2.3 pb_frontend interface

Interface slouží k ovládání dma jednotky pomocí 8-bitových portů, jako například při připojení k PicoBlaze mikrokontroléru. V axi_pb_dma má pb_frontend 8 kanálů. Každý kanál má 256x8bit paměť uspořádanou následujícím způsobem:

Adresa

0x00 - 0x3F	- 8x r/w channel configuration, 8 byte per channel
0x4x - 0x7x	- reserved for firmware operations
0x80 - 0x87	- Commands - wr only, initiates DMA transfer
0xC0	- Status - rd only, busy bits for each dma channel

3. Použité/Potřebné vybavení a nástroje

Xilinx Design Suite 14.3 (2012), Xilinx development board ZC702 (Zynq)

4. Jak spustit

- Otevřete přiložený PlanAhead projekt.
- Použijte iMPACT pro naprogramování FPGA
- Spusťte SDK (File->Export->Export hardware for SDK ...)
- Vyberte jeden ze tří projektů a spusťte ho obvyklým způsobem

5. Známé problémy a jejich odstranění

Obsah CD je třeba zkopírovat do cesty bez mezer. V opačném případě nemusí PlanAhead pracovat správně.

6. Obsah a popis přiloženého balíku

Struktura balíčku

cdrom - 14_3_test_axi_pb_dma
 - doc
 - include

PlanAhead projekt s testovacími designy
tato dokumentace
programování axi_pb_dma přes axi-lite