

Mikroračunarski sistemi za rad u realnom vremenu

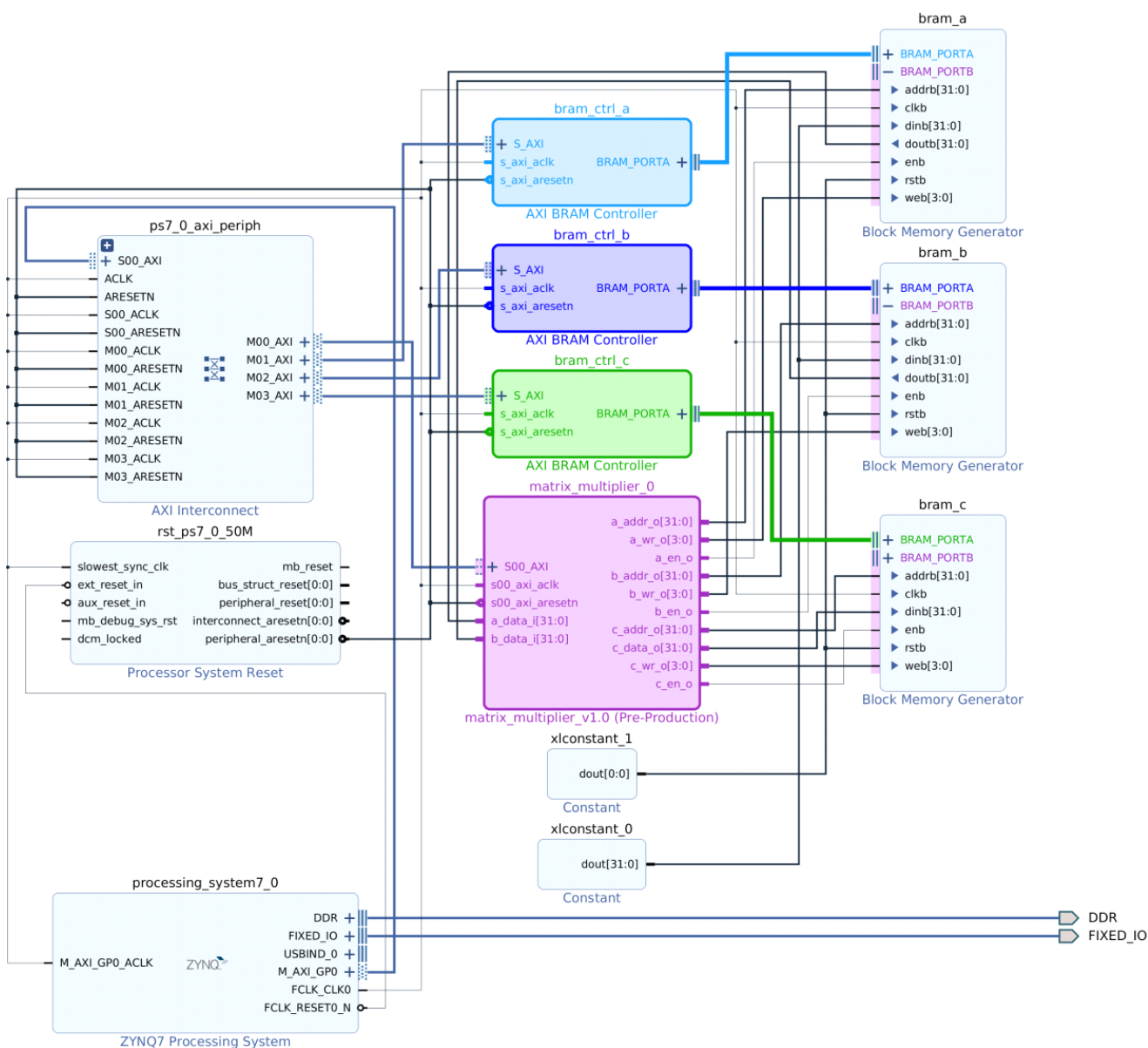
Projekat 23.1.1

Množać matrica (Matrix Multiplier)

2024 Izborni

Vesna Jankovic EE187/2019

Tekst zadatka projekta: Razviti Linux drajver i odgovarajuću aplikaciju koja koristi drajver i demonstrira kontrolu i upravljanje MatrixMultiplieP jezgrom.



Opis strukture hardvera: MatrixMultiplierP jezgro (roza boja na blok dijagramu) je memorijski mapirano u adresni prostor ZYNQ procesora. Ono služi za računanje matričnog proizvoda dve matrice ispunjene pozitivnim celim brojevima. Dve matrice nad kojima se računa proizvod su smeštene u BRAM memorije " bram_đ" i " bram_đ", dok se proizvod upisuje u BRAM memoriju " bram_ć". BRAM memorije su dvopristupne, što znači da imaju dva interfejsa za pristup podacima (portA i portB). MatrixMultiplierkoristi portBpristupe kako bi čitao iz " bram_đ" i " bram_đ" memorija i nakon računanja proizvoda, upisao u memoriju bram_ć. BRAM kontrolera " bram_ctr_đ (plava), " bram_ctr_đ (ljubicasta) i " bram_ctr_ć (zelena boja) može da upisuje ili čita vrednosti iz BRAM memorija preko njihovog drugog pristupa - "portA". Procesor takođe može da konfiguriše i zadaje komande MatrixMultiplier jezgru preko AXI Lite interfejsa.

$$\begin{matrix} & \mathbf{A_{2 \times 3}} & & \mathbf{B_{3 \times 4}} & & \mathbf{C_{2 \times 4}} \\ \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} & \times & \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix} & = & \begin{bmatrix} 38 & 44 & 50 & 56 \\ 83 & 98 & 113 & 128 \end{bmatrix}
 \end{matrix}$$

bram_a		bram_b		bram_c	
0	1	0	1	0	38
4	2	4	2	4	44
8	3	8	3	8	50
12	4	12	4	12	56
16	5	16	5	16	83
20	6	20	6	20	98
24	-	24	7	24	113
28	-	28	8	28	128
32	-	32	9	32	-
36	-	36	10	36	-
40	-	40	11	40	-
44	-	44	12	44	-
48	-	48	-	48	-
...

Pri množenju matrice A (dimenzija n x m) i matrice B (dimenzija m x p) dobija se matrica C (dimenzija n x p). MatrixMultiplier poseduje registre "n", "m" i "p" u koje je potrebno upisati dimenzije matrica nad kojima se računa proizvod. Maksimalna vrednost ovih registara j 7, tako da je najveći proizvod koji je moguće izračunati pomoću ovog jezgra $A_{7 \times 7} \times B_{7 \times 7} = C_{7 \times 7}$. Sem ova tri registra, MatrixMultiplierjoš ima dva registra: statusni registar "ready" i kontrolni registar "start". Prvi govori da li je MatrixMultiplierzavršio sa računanjem prethodnog proizvoda i da li je spreman za novi račun. Start se koristi kako bi se pokrenulo računanje (potrebno ga je postaviti na jedinicu a neposredno nakon toga na nulu). Na sledecoj slici je dat primer množenja matrica. Za date matrice se može videti kakav treba biti raspored elemenata matrice po adresama u bram_a i bram_b memorijama, te kako će rešenje biti smešteno u bram_c Kao što se može primetiti, elementi matrica se smeštaju red po red, jedan za drugim u BRAM memorije. Bitno je pomenuti, da iako su lokacije u BRAM memorijama 32-bitne, elementi matrica A i B ne trebaju biti veći od 12 bita, tj maksimalna vrednost elementa matrice je $2^{12}=4096$.

Tipičan rad sa MatrixMultiplier modulom je:

- Pomoću procesora upisati matricu A u bram_a a matricu B u bram_b
- Podesiti registre n,m i p u MatrixMultipliemodulu u zavisnosti od dimenzija matrica A i B.
- Pokrenuti MatrixMultiplierupisom 1 a zatim 0 u startregistar.
- Periodično čitati ready registar, kada se postavi na 1 znači da je modul završio računanje
- Pomoću procesora pročitati rezultat množenja matrica iz bram_cmemorije

Lokacijama BRAM memorija se pristupa direktno, pomoću bazne adrese bram_ctrl modula i ofseta za željenu adresu. Kao što se vidi sa prethodne slike, adresa sledećeg podatka je uvek pomerena za 4 u odnosu na prethodnu.

Registrima MatrixMultiplier jezgra se pristupa direktno, pomoću bazne adrese i ofseta za željeni registar.

Offset	Registar
0	ready
4	start
8	n
12	m
16	p

1. Koristeći uputstvo koje se može pronaći na stranici predmeta, pripremiti SD karticu sa svim potrebnim datotekama (BOOT.bin, devicetree.dtb i ulmage) koje omogućavaju podizanje Linux operativnog sistema, sa gore opisanim dizajnom uključenim u BOOT.bin datoteku

2. Napisati Linux drajver za MatrixMultipliemodul sa sledećom funkcionalnostima:

- Drajver dobija slobodne upravljačke brojeve (MAJOR, MINOR) od operativnog sistema. Potrebno je pri pokretanju insmod funkcije, automatski kreirati ukupno četiri node fajla u /dev direktorijumu pri čemu svi imaju isti MAJOR a različite MINOR brojeve:
 - /dev/matmul (za komunikaciju sa MatrixMultiplierP blokom)
 - /dev/bram_a (za upis matrice A u bram_amemoriju)
 - /dev/bram_b (za upis matrice B u bram_bmemoriju)
 - /dev/bram_c (za čitanje matrice C iz bram_cmemorije)

- Čitanje vrednosti registara iz IP bloka vrši se pomoću readsistemskog poziva sa datotekom uređaja /dev/ matmul na sledeći način:

```
$ cat /dev/matmul
```

Nakon ove komande trebalo bi da se ispišu vrednosti svih registara u sledecem formatu:

```
ready=?;start=?;n=?;m=?p=?
```

gde ? predstavlja trenutnu vrednost registra.

- Upis u registre MatrixMultiplier:

```
$ echo "dim=2,3,4;"
```

- Postavlja registre n,m i p za dimenzije matrica na željene vrednosti

(ograničiti da je maksimalna vrednost ovih parametara 7)

```
$ echo "start=1" > /dev/matmul
```

- postavlja start registar na 1

```
$ echo "start=0" > /dev/matmul
```

- postavlja start registar na 0

```
$ echo "start=trigger " > /dev/matmul
```

-postavlja start na 1 a odmah nakon toga na 0

- Upis matrice u /dev/bram_a ili /dev/bram_b

Matrice se upisuju tako što se prosledi kompletna matrica kao string. Elementi se odvajaju zarezom a redovi tačka-zarezom. Na primer, kako bi se upisala matrica A iz primera sa slike:

```
echo "1,2,3;4,5,6;" > /dev/bram_a
```

Pri upisu je potrebno proveriti da li je data validna matrica, te u suprotnom da ispiše poruku upozorenja. Na primer matrica "1,2;4,5,6,7;" nije validna jer u prvom redu ima 2 elementa a drugom 4.

- Čitanje rezultata (matrice) iz /dev/bram_c

```
cat /dev/bram_c
```

ispisuje rezultat u prethodno pomenutom formatu. Za primer sa slike, to bi bilo:

```
38,44,50,56;83,98,113,128;
```