Lab #9: Direct Memory Access

05/31/2018

4190.309A: Hardware System Design
(Spring 2018)

Notations

- HOST$ XXX
  - Type XXX at the terminal of your Ubuntu-PC.
- BOARD$ YYY
  - Type YYY at the terminal of ZedBoard (on minicom).
- TCL$ ZZZ
  - Type ZZZ at the Tcl console of Vivado.

Overview

- Step-by-step procedure: Apply DMA to an example design
  - Download source codes
    - See main.c
  - Implementation and its performance analysis
    - Create new project for ZedBoard and apply scripts for a new block design.
    - Generate a bitstream & replace the bootloader (u-boot.img.nc → u-boot.img)
    - Put the generated bitstream and all files in “HSD_LAB9” into SD card .
    - Make and run with checking performance

- Get the source code before get into the practice
  - `HOST$ git clone git://147.47.208.211/HSD_LAB9.git`

Simulation Verification

Performance Analysis

```
gettimeofday (&start, NULL);
// compute on cpu
for (i = 0; i < SIZE; i++)
  for (j = 0; j < SIZE; j++)
    output[i] +=
        input[SIZE-1-j] * mat[SIZE-1-j][i];
gettimeofday (&end, NULL);
```

-> 409us

Performance Analysis

```
gettimeofday (&start, NULL);
// load to bram
for (i = 0; i < SIZE * (SIZE + 1); i++)
  *(fpga_bram + i) = flat[i];
gettimeofday (&end, NULL);
```

-> 876us

```
gettimeofday (&start, NULL);
// wait for IP
while (*fpga_ip == 0x5555);
gettimeofday (&end, NULL);
```

-> 95us

```
gettimeofday (&start, NULL);
//get result
for (i = 0; i < SIZE; i++)
  output_fpga[i] = *(fpga_bram + i);
gettimeofday (&end, NULL);
```

-> 14us

ARM-CPU (409us) < FPGA offloading (985us)

Performance Tuning : MEMCPY

- memcpy: copy block of memory
  - `void* memcpy (void* dest, const void* src, size_t num);`
    - dest: pointer to the destination array
    - src: pointer to the source of data to be copied
    - num: number of bytes to copy

```
gettimeofday (&start, NULL);
// load to bram
for (i = 0; i < SIZE * (SIZE + 1); i++)
  *(fpga_bram + i) = flat[i];
gettimeofday (&end, NULL);
```

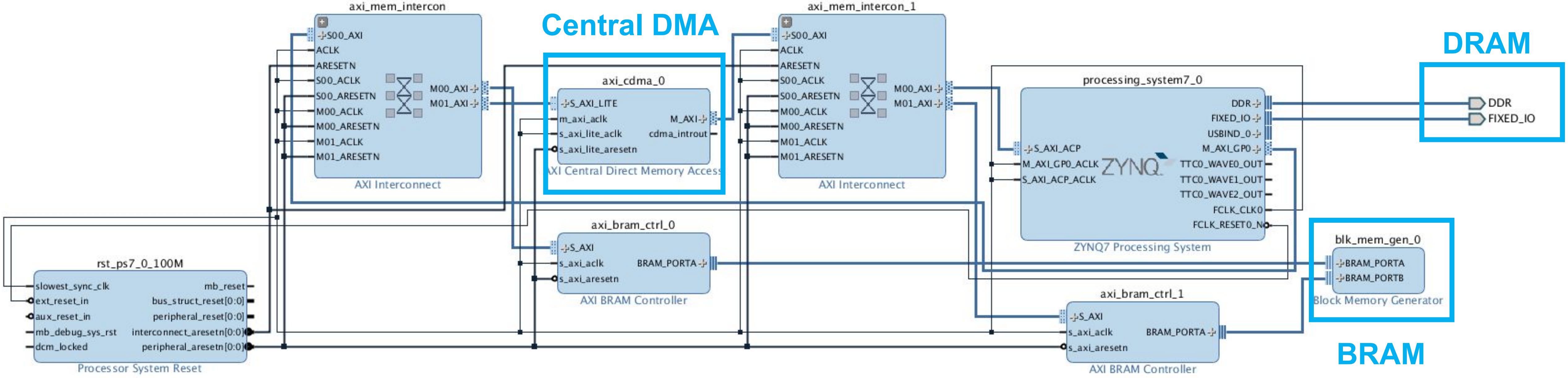
-> 876us

```
gettimeofday (&start, NULL);
// memcpy to bram
size_t size = SIZE * (SIZE + 1) * sizeof(uint32_t);
memcpy( fpga_bram, flat, size);
gettimeofday (&end, NULL);
```

-> ?

Performance Tuning : DMA

- Create a Vivado project and run block_design_dma.tcl
  - TCL$ source block_design_dma.tcl



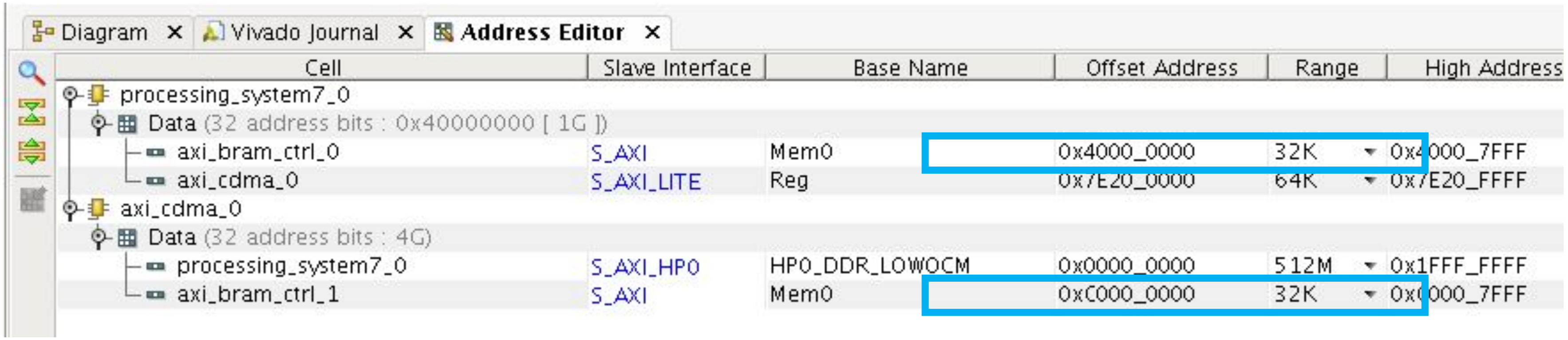
Central DMA

DRAM

BRAM

Performance Tuning : DMA

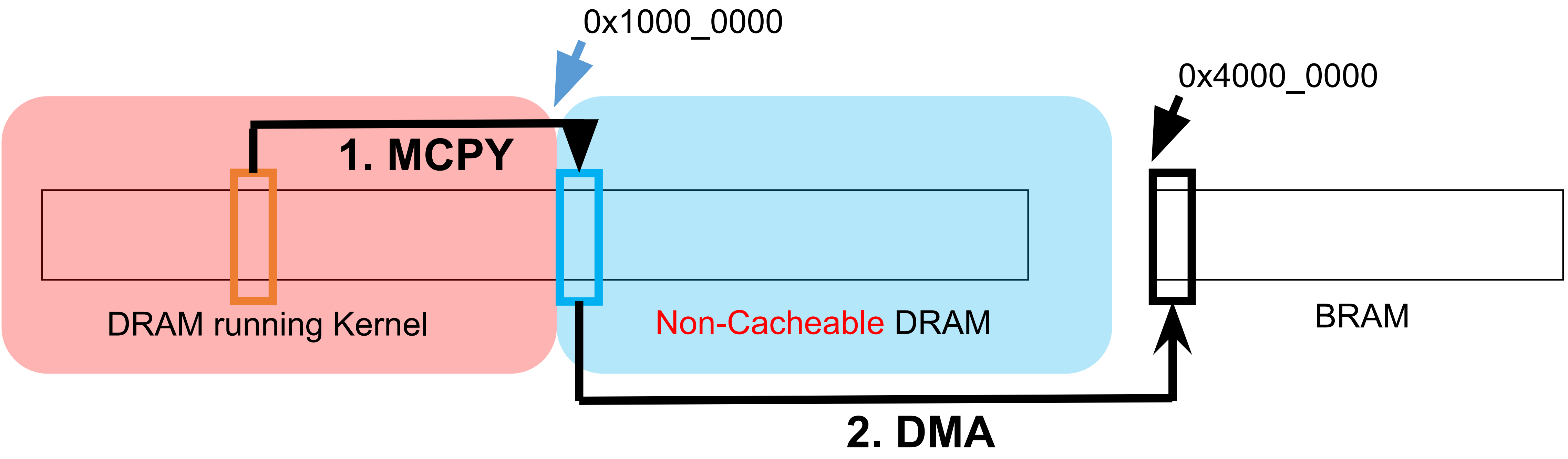
- BRAM can be either accessed from processing system (0x4000_0000) or accessed from the cdma module (0xC000_0000)
  - See how address is mapped for DMA IP and BRAM



Cell	Slave Interface	Base Name	Offset Address	Range	High Address
processing_system7_0					
Data (32 address bits : 0x40000000 [1G])					
axi_bram_ctrl_0	S_AXI	Mem0	0x4000_0000	32K	0x4000_7FFF
axi_cdma_0	S_AXI_LITE	Reg	0x7E20_0000	64K	0x7E20_FFFF
axi_cdma_0					
Data (32 address bits : 4G)					
processing_system7_0	S_AXI_HP0	HP0_DDR_LOWOCM	0x0000_0000	512M	0x1FFF_FFFF
axi_bram_ctrl_1	S_AXI	Mem0	0xC000_0000	32K	0xC000_7FFF

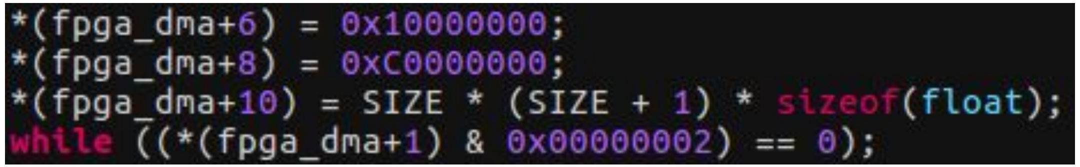
Performance Tuning : DMA

- Update the boot loader to enable non-cacheable DRAM
  - Replace u-boot.img with the newly provided one (*u-boot.img.nc*)
  - **CAUTION**: Use *u-boot.img.nc* only when you want to enable DMA



Performance Tuning : DMA

- To perform DMA operation, follow below :



```
*(fpga_dma+6) = 0x10000000;
*(fpga_dma+8) = 0xC0000000;
*(fpga_dma+10) = SIZE * (SIZE + 1) * sizeof(float);
while ((*(fpga_dma+1) & 0x00000002) == 0);
```

  - Assign source address
  - Assign destination address
  - Assign the number of bytes to transfer
  - Poll if the operations is done

- Refer to *AXI Central Direct Memory Access p. 13* for details
  - HSD_LAB9/pg034-axi-cdma.pdf

Today’s Todo

1. Create new project for ZedBoard and apply new block design.
2. Generate a bitstream.
3. Connect SD Card and replace bootloader.
4. Put the new bitstream, C files and input file into the SD Card.
5. Run the C program on new bitstream so we can check the performance of memcpy and DMA.

Grading policy

- Check list
  - C output on FPGA, with memcpy and DMA performance (100 points)

- Submit “L9.**pdf**” (containing a screenshot only) on eTL
  - Due : 6/5 (Tue) 11:59 PM