

# Hardware/Software Codesign Lab 4

Student Name: Kuldeep Gohil

Student ID: 015499534

1. Follow the Lab 4 manual finish Lab 4.
2. Copy and paste the following information to the end of this document and submit this document:
  - 1) system.mss: highlight information for the custom IP added and BRAM and BRAM controller.

## Peripheral Drivers

Drivers present in the Board Support Package.

axi_bram_ctrl_0	bram	<a href="#">Documentation</a>	<a href="#">Import Examples</a>
buttons	gpio	<a href="#">Documentation</a>	<a href="#">Import Examples</a>
led_ip	led_ip		
ps7_afi_0	generic		
ps7_afi_1	generic		
ps7_afi_2	generic		
ps7_afi_3	generic		
ps7_coresight_comp_0	coresightps_dcc	<a href="#">Documentation</a>	
ps7_dds_0	ddrps	<a href="#">Documentation</a>	
ps7_dds_0	generic		
ps7_dev_cfg_0	devcfg	<a href="#">Documentation</a>	<a href="#">Import Examples</a>
ps7_dma_ns	dmaps	<a href="#">Documentation</a>	<a href="#">Import Examples</a>
ps7_dma_s	dmaps	<a href="#">Documentation</a>	<a href="#">Import Examples</a>
ps7_globaltimer_0	generic		
ps7_gpv_0	generic		
ps7_intc_dist_0	generic		
ps7_iop_bus_config_0	generic		
ps7_l2cachec_0	generic		
ps7_ocmc_0	generic		
ps7_pl310_0	generic		
ps7_pmu_0	generic		
ps7_ram_0	generic		
ps7_ram_1	generic		
ps7_scuc_0	generic		
ps7_scugic_0	scugic	<a href="#">Documentation</a>	<a href="#">Import Examples</a>
ps7_scutimer_0	scutimer	<a href="#">Documentation</a>	<a href="#">Import Examples</a>
ps7_scuwdt_0	scuwdt	<a href="#">Documentation</a>	<a href="#">Import Examples</a>
ps7_slcr_0	generic		
ps7_uart_1	uartps	<a href="#">Documentation</a>	<a href="#">Import Examples</a>
ps7_xadc_0	xadcps	<a href="#">Documentation</a>	<a href="#">Import Examples</a>
switches	gpio	<a href="#">Documentation</a>	<a href="#">Import Examples</a>

- 2) Memory dump information for two cases: Case 1: all four sections of executable are in DDR3; Case 2. code and data in DDR3, stack and heap in BRAM.

Case 1:

```
lab4.elf:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text          00001a04 00100000 00100000 00010000 2**6
    CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .init           00000018 00101a04 00101a04 00011a04 2**2
    CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .fini           00000018 00101a1c 00101a1c 00011a1c 2**2
    CONTENTS, ALLOC, LOAD, READONLY, CODE
  3 .rodata         0000018c 00101a34 00101a34 00011a34 2**2
    CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .data           00000498 00101bc0 00101bc0 00011bc0 2**3
    CONTENTS, ALLOC, LOAD, DATA
  5 .eh_frame       00000004 00102058 00102058 00012058 2**2
    CONTENTS, ALLOC, LOAD, READONLY, DATA
  6 .mmu_tbl        00004000 00104000 00104000 00014000 2**0
    CONTENTS, ALLOC, LOAD, READONLY, DATA
  7 .init_array     00000004 00108000 00108000 00018000 2**2
    CONTENTS, ALLOC, LOAD, DATA
  8 .fini_array     00000004 00108004 00108004 00018004 2**2
    CONTENTS, ALLOC, LOAD, DATA
  9 .ARM.attributes 00000033 00108008 00108008 00018008 2**0
    CONTENTS, READONLY
10 .bss             00000030 00108008 00108008 00018008 2**2
    ALLOC
11 .heap            00000408 00108038 00108038 00018008 2**0
    ALLOC
12 .stack           00001c00 00108440 00108440 00018008 2**0
    ALLOC
13 .comment         00000031 00000000 00000000 0001803b 2**0
    CONTENTS, READONLY
14 .debug_info      000063b6 00000000 00000000 0001806c 2**0
    CONTENTS, READONLY, DEBUGGING
15 .debug_abbrev    00001709 00000000 00000000 0001e422 2**0
    CONTENTS, READONLY, DEBUGGING
16 .debug_aranges   000001f8 00000000 00000000 0001fb30 2**3
    CONTENTS, READONLY, DEBUGGING
17 .debug_macro     00002f9f 00000000 00000000 0001fd28 2**0
    CONTENTS, READONLY, DEBUGGING
18 .debug_line      00001e83 00000000 00000000 00022cc7 2**0
    CONTENTS, READONLY, DEBUGGING
19 .debug_str        0000e617 00000000 00000000 00024b4a 2**0
    CONTENTS, READONLY, DEBUGGING
20 .debug_frame     000003fc 00000000 00000000 00033164 2**2
    CONTENTS, READONLY, DEBUGGING
21 .debug_loc       000011c1 00000000 00000000 00033560 2**0
    CONTENTS, READONLY, DEBUGGING
22 .debug_ranges    00000188 00000000 00000000 00034721 2**0
    CONTENTS, READONLY, DEBUGGING
```

Case 2:

```
lab4.elf:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA           LMA           File off  Algn
  0 .text          00001a04  00100000  00100000  00010000  2**6
    CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .init           00000018  00101a04  00101a04  00011a04  2**2
    CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .fini           00000018  00101a1c  00101a1c  00011a1c  2**2
    CONTENTS, ALLOC, LOAD, READONLY, CODE
  3 .rodata         0000018c  00101a34  00101a34  00011a34  2**2
    CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .data           00000498  00101bc0  00101bc0  00011bc0  2**3
    CONTENTS, ALLOC, LOAD, DATA
  5 .eh_frame       00000004  00102058  00102058  00012058  2**2
    CONTENTS, ALLOC, LOAD, READONLY, DATA
  6 .mmu_tbl        00004000  00104000  00104000  00014000  2**0
    CONTENTS, ALLOC, LOAD, READONLY, DATA
  7 .init_array     00000004  00108000  00108000  00018000  2**2
    CONTENTS, ALLOC, LOAD, DATA
  8 .fini_array     00000004  00108004  00108004  00018004  2**2
    CONTENTS, ALLOC, LOAD, DATA
  9 .ARM.attributes 00000033  00108008  00108008  00018008  2**0
    CONTENTS, READONLY
10 .bss             00000030  00108008  00108008  00018008  2**2
    ALLOC
11 .heap            00000400  40000000  40000000  00020000  2**0
    ALLOC
12 .stack           00001c00  40000400  40000400  00020000  2**0
    ALLOC
13 .comment         00000031  00000000  00000000  0001803b  2**0
    CONTENTS, READONLY
14 .debug_info      000063b6  00000000  00000000  0001806c  2**0
    CONTENTS, READONLY, DEBUGGING
15 .debug_abbrev    00001709  00000000  00000000  0001e422  2**0
    CONTENTS, READONLY, DEBUGGING
16 .debug_aranges   000001f8  00000000  00000000  0001fb30  2**3
    CONTENTS, READONLY, DEBUGGING
17 .debug_macro     00002f9f  00000000  00000000  0001fd28  2**0
    CONTENTS, READONLY, DEBUGGING
18 .debug_line      00001e83  00000000  00000000  00022cc7  2**0
    CONTENTS, READONLY, DEBUGGING
19 .debug_str       0000e617  00000000  00000000  00024b4a  2**0
    CONTENTS, READONLY, DEBUGGING
20 .debug_frame     000003fc  00000000  00000000  00033164  2**2
    CONTENTS, READONLY, DEBUGGING
21 .debug_loc       000011c1  00000000  00000000  00033560  2**0
    CONTENTS, READONLY, DEBUGGING
22 .debug_ranges    00000188  00000000  00000000  00034721  2**0
    CONTENTS, READONLY, DEBUGGING
```

3. Answer the following question:

- 1) Specify the location(s) for the DDR3 Controller and DDR3 memory: inside Xilinx Zynq-7000 (XC7Z010-1CLG400C) or outside? If inside, specify if it is a hard core or soft cores and where does it locate in XC7Z010-1CLG400C: in PS or PL.

DDR3 Controller is inside Zynq PS, specifically hard core. DDR3 Memory is outside Zynq.

- 2) Specify the location(s) for the AXI-BRAM Controller and BRAM in this lab: inside Xilinx Zynq-7000 (XC7Z010-1CLG400C) or outside? If inside, specify if it is a hard core or soft cores and where does it locate in XC7Z010-1CLG400C: in PS or PL.

AXI-BRAM Controller and BRAM in this lab are both inside Zynq PL, specifically soft core.

- 3) Specify the locations assigned to the code, data, stack and heap section of your software executable for the two linker script settings tested in the lab.

First it was assigned in DDR3 Memory and then it was in BRAM.

- 4) List all the external peripherals in the embedded system you build in this lab.

Push buttons, dip switches, LEDs, and DDR3 memory UART.