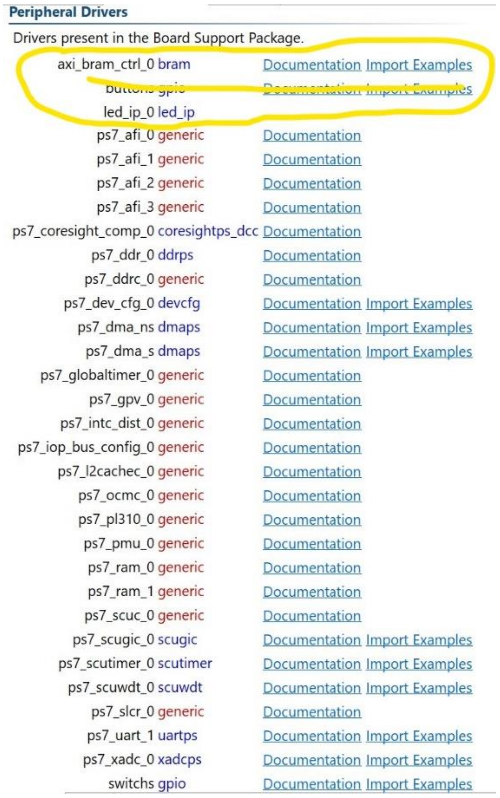
Hardware/Software Codesign Lab 4

Student Name: Lauro Cabral

Student ID: 012733966

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.

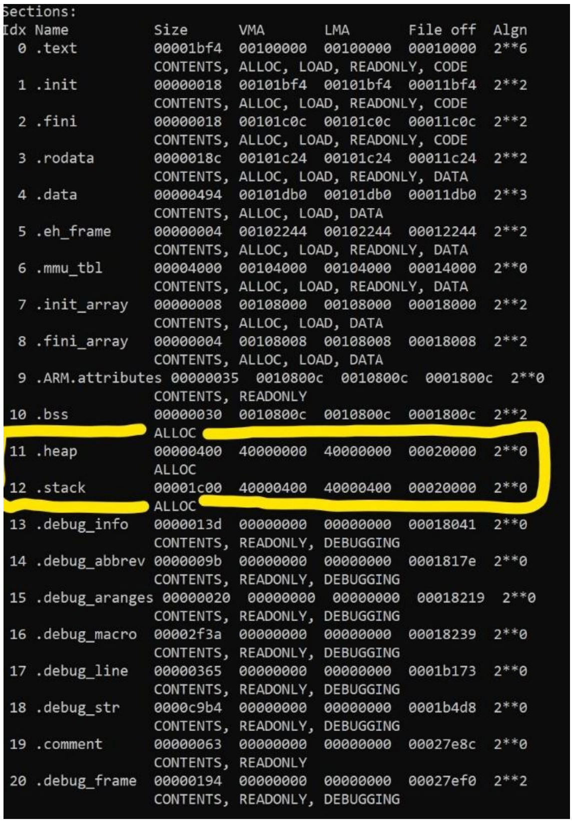


Peripheral Drivers

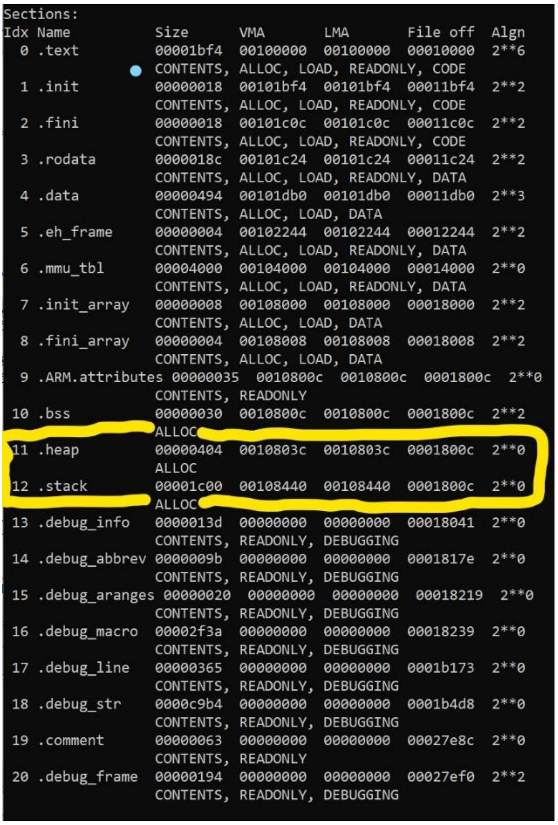
Drivers present in the Board Support Package.

axi_bram_ctrl_0	bram	Documentation Import Examples
buttons	gpio	Documentation Import Examples
led_ip_0	led_ip	
ps7_afi_0	generic	Documentation
ps7_afi_1	generic	Documentation
ps7_afi_2	generic	Documentation
ps7_afi_3	generic	Documentation
ps7_coresight_comp_0	coresightps_dcc	Documentation
ps7_ddr_0	ddrps	Documentation
ps7_ddrc_0	generic	Documentation
ps7_dev_cfg_0	devcfg	Documentation Import Examples
ps7_dma_ns	dmaps	Documentation Import Examples
ps7_dma_s	dmaps	Documentation Import Examples
ps7_globaltimer_0	generic	Documentation
ps7_gpv_0	generic	Documentation
ps7_intc_dist_0	generic	Documentation
ps7_iop_bus_config_0	generic	Documentation
ps7_l2cachec_0	generic	Documentation
ps7_ocmc_0	generic	Documentation
ps7_pl310_0	generic	Documentation
ps7_pmu_0	generic	Documentation
ps7_ram_0	generic	Documentation
ps7_ram_1	generic	Documentation
ps7_scuc_0	generic	Documentation
ps7_scugic_0	scugic	Documentation Import Examples
ps7_scutimer_0	scutimer	Documentation Import Examples
ps7_scuwdt_0	scuwdt	Documentation Import Examples
ps7_slcr_0	generic	Documentation
ps7_uart_1	uartps	Documentation Import Examples
ps7_xadc_0	xadcps	Documentation Import Examples
switchs	gpio	Documentation Import Examples

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.



```
Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00001bf4  00100000  00100000  00010000  2**6
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .init         00000018  00101bf4  00101bf4  00011bf4  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .fini         00000018  00101c0c  00101c0c  00011c0c  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  3 .rodata       0000018c  00101c24  00101c24  00011c24  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .data         00000494  00101db0  00101db0  00011db0  2**3
                  CONTENTS, ALLOC, LOAD, DATA
  5 .eh_frame     00000004  00102244  00102244  00012244  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  6 .mmu_tbl      00004000  00104000  00104000  00014000  2**0
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  7 .init_array   00000008  00108000  00108000  00018000  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  8 .fini_array   00000004  00108008  00108008  00018008  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  9 .ARM.attributes 00000035  0010800c  0010800c  0001800c  2**0
                  CONTENTS, READONLY
 10 .bss          00000030  0010800c  0010800c  0001800c  2**2
                  ALLOC
 11 .heap         00000400  40000000  40000000  00020000  2**0
                  ALLOC
 12 .stack        00001c00  40000400  40000400  00020000  2**0
                  ALLOC
 13 .debug_info   0000013d  00000000  00000000  00018041  2**0
                  CONTENTS, READONLY, DEBUGGING
 14 .debug_abbrev 0000009b  00000000  00000000  0001817e  2**0
                  CONTENTS, READONLY, DEBUGGING
 15 .debug_aranges 00000020  00000000  00000000  00018219  2**0
                  CONTENTS, READONLY, DEBUGGING
 16 .debug_macro  00002f3a  00000000  00000000  00018239  2**0
                  CONTENTS, READONLY, DEBUGGING
 17 .debug_line   00000365  00000000  00000000  0001b173  2**0
                  CONTENTS, READONLY, DEBUGGING
 18 .debug_str    0000c9b4  00000000  00000000  0001b4d8  2**0
                  CONTENTS, READONLY, DEBUGGING
 19 .comment      00000063  00000000  00000000  00027e8c  2**0
                  CONTENTS, READONLY
 20 .debug_frame  00000194  00000000  00000000  00027ef0  2**2
                  CONTENTS, READONLY, DEBUGGING
```



```
Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00001bf4  00100000  00100000  00010000  2**6
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .init         00000018  00101bf4  00101bf4  00011bf4  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .fini         00000018  00101c0c  00101c0c  00011c0c  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  3 .rodata       0000018c  00101c24  00101c24  00011c24  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .data         00000494  00101db0  00101db0  00011db0  2**3
                  CONTENTS, ALLOC, LOAD, DATA
  5 .eh_frame     00000004  00102244  00102244  00012244  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  6 .mmu_tbl      00004000  00104000  00104000  00014000  2**0
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  7 .init_array   00000008  00108000  00108000  00018000  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  8 .fini_array   00000004  00108008  00108008  00018008  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  9 .ARM.attributes 00000035  0010800c  0010800c  0001800c  2**0
                  CONTENTS, READONLY
 10 .bss          00000030  0010800c  0010800c  0001800c  2**2
                  ALLOC
 11 .heap         00000404  0010803c  0010803c  0001800c  2**0
                  ALLOC
 12 .stack        00001c00  00108440  00108440  0001800c  2**0
                  ALLOC
 13 .debug_info   0000013d  00000000  00000000  00018041  2**0
                  CONTENTS, READONLY, DEBUGGING
 14 .debug_abbrev 0000009b  00000000  00000000  0001817e  2**0
                  CONTENTS, READONLY, DEBUGGING
 15 .debug_aranges 00000020  00000000  00000000  00018219  2**0
                  CONTENTS, READONLY, DEBUGGING
 16 .debug_macro  00002f3a  00000000  00000000  00018239  2**0
                  CONTENTS, READONLY, DEBUGGING
 17 .debug_line   00000365  00000000  00000000  0001b173  2**0
                  CONTENTS, READONLY, DEBUGGING
 18 .debug_str    0000c9b4  00000000  00000000  0001b4d8  2**0
                  CONTENTS, READONLY, DEBUGGING
 19 .comment      00000063  00000000  00000000  00027e8c  2**0
                  CONTENTS, READONLY
 20 .debug_frame  00000194  00000000  00000000  00027ef0  2**2
                  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.

      The DDR memory is connected to the same AXI bus infrastructure that is used by the ARMs CPUs (PS or Processor Subsystem) - but it is on the "far side" from the AXI interfaces onto the FPGA fabric. So, to use the DDR in a pure FPGA design you still have configured the PS, which configures whole AXI bus infrastructure, then tell the CPUs to do nothing. DDR3 is outside of 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.

      In Zynq, BRAM is available in the FPGA PL as a softcore. We need to create a BRAM controller soft core to control the access of the BRAM 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 in DDR3 memory, second in BRAM

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

      LEDs, Push Button, Dip Switch , DDR3 Memory , UART