**********DRAFT**********
SDRAM
11/01/18
**********DRAFT**********

For the ICOBoard you did not use PLL due hardware issues(PLL & SRAM) would a PLL be best for the catzip & sdram? Looking at the DCM lines 116 to 122 of xulalx25soc?
Starting looking at how the xulalx25soc intergrates the sdram.

Sdram catzip xulalx25soc

- Autofpga project sdramdev.txt
- Catzip HX8K FPGA
- 50 MHz refresh_clk <= 10'd391
- wbsdram.v wishbone interface & sdramcontroller
- Verilator simulation arm-main_tb compiled with sdramsim.cpp & sdramsim.h
- Wishbone 32 bit addr & 32 bit data
- 0x2000000 to 0x2FFFFFF
- Write ./arm-wbregs 0x2000004 0x55aaaa55
- toplevel.v
  - main.v
    - Wbsdram.v

- Non Autofpga project maybe soon
- Xulalx25soc LX25 or LX9 FPGA
- 80 MHz refresh_clk <= 10'd625;
- wbsdram.v wishbone interface & sdramcontroller
- Verilator simulation busmaster_tb compiled with sdramsim.cpp & sdramsim.h
- Wishbone 32 bit addr & 32 bit data
- 0x2000000 to 0x2FFFFFF
- Write ./wbregs 0x2000004 0x55aaaa55
- Read ./wbregs 0x2000004
- toplevel.v
  - busmaster,v
    - Wbsdram.v

Using autofpga to add sdram & sdramscope support to catzip
Both the xula2-lx25 & xula2-lx9 & catzip use the same sdram IC.
The xulalx25soc runs on the xula2-lx25 & xula2-lx9 and the has a verilator simulation. The xulalx25soc design has a sdram C++ simulator implemented in the files sdramsim.cpp & sdramsim.h. In addition the xulalx25soc design has the wbsdram.v file which currently builds the toplevel.bit for the xula2-lx9 or xula2-lx25.

Create the sdramdev.txt file to be used by autodata to include in the toplevel.v & main.v which is instaniated by toplevel.v

This is what goes in toplevel.v

```
module  toplevel(i_clk,
                i_ram_feedback_clk, o_ram_clk, o_ram_cke, o_ram_cs_n, o_ram_ras_n, o_ram_cas_n,
                o_ram_we_n, o_ram_bs, o_ram_addr, o_ram_udqm, o_ram_ldqm,
                io_ram_data,
```

```
  //o_ram_cs_n,     Chip select
  //o_ram_cke,      Clock enable
  //o_ram_ras_n,    Row address strobe
  //o_ram_cas_n,    Column address strobe
  //o_ram_we_n,     Write enable
  //o_ram_bs,       Bank select
  //o_ram_addr,     Address lines
  //r_ram_data,     Data lines (input)
  //ram_data,       Data lines (output)
  input          i_ram_feedback_clk;
  output  wire    o_ram_clk, o_ram_cke;
  output  wire    o_ram_cs_n, o_ram_ras_n, o_ram_cas_n, o_ram_we_n;
  output  wire    [1:0]   o_ram_bs;
  output  wire    [12:0]  o_ram_addr;
  output  wire           o_ram_udqm, o_ram_ldqm;
  inout   wire    [15:0]  io_ram_data;
  wire    [15:0]  ram_data;
  wire           ram_drive_data;
  //wire         o_ram_drive_data;
  reg     [15:0]  r_ram_data;

  reg     [1:0]   o_ram_dqm;
  //using instead { o_ram_udqm, o_ram_ldqm }
  wire    [31:0]  o_debug;
                        .
                        .
                        .
 main    thedesign(s_clk, s_reset,
        //*********************************************************
        //This section was missing when the simulation was aborting
        //xulalx25soc/rtl/busmaster.v same as main.v in new design
        //o_ram_cs_n, o_ram_cke, o_ram_ras_n, o_ram_cas_n,
        //o_ram_we_n, o_ram_bs, o_ram_addr,
        //o_ram_drive_data, i_ram_data, o_ram_data,
        //o_ram_dqm,
        //o_ram_drive seems to be the same as ram_drive_data
        //o_ram_drive_data was added to SIM.TICK and the assert(driv);
        //sdramsim.cpp restored write
        //assert(!driv); sdramsim.cpp restored read
        //*********************************************************

        o_ram_cs_n, o_ram_cke, o_ram_ras_n, o_ram_cas_n, o_ram_we_n,
              o_ram_bs, o_ram_addr,
              ram_drive_data, i_ram_data, o_ram_data, { o_ram_udqm, o_ram_ldqm },
        o_debug
,
        // GPIO wires
                        .
                        .
                        .
 // SDRAM Interface
  //
```

```
        // Use the PPIO primitive to give us access to a group of SB_IO's,
        // and therefore access to a tristate output
        //
        ppio #(.W(16))
                sdramioi(o_ram_we_n, io_ram_data, o_ram_data, i_ram_data);
assign o_ram_clk = clk_50mhz;
```

This is what goes in main.v

```
`ifdef  SDRAM_ACCESS
wbsdram sdrami(i_clk,
                wb_cyc, (wb_stb)&&(sdram_sel),
                /* verilator lint_off WIDTH */
                wb_we, wb_addr[(25-3):0], wb_data, wb_sel,
                /* verilator lint_off WIDTH */
                sdram_ack, sdram_stall, sdram_data,
                o_ram_cs_n, o_ram_cke, o_ram_ras_n, o_ram_cas_n, o_ram_we_n,
                o_ram_bs, o_ram_addr,
                ram_drive_data, i_ram_data, o_ram_data, o_ram_dqm,
                o_debug);

`else   // SDRAM_ACCESS

        // In the case that there is no sdram peripheral responding on the wb bus
        reg     r_sdram_ack;
        initial r_sdram_ack = 1'b0;
        always @(posedge i_clk) r_sdram_ack <= (wb_stb)&&(sdram_sel);
        assign  sdram_ack   = r_sdram_ack;
        assign  sdram_stall = 0;
        assign  sdram_data  = 0;

`endif  // SDRAM_ACCESS
```

Adding a scope to debug the sdram issue
Create the sdramscope.txt file to be used by autodata to include
in the main.v which is called by toplevel.v

This is what goes in main.v

```
        assign  sdram_debug = { (!o_ram_ce_n),
                        wb_cyc, (wb_stb)&&(sdram_sel), wb_we,
                                sdram_stall,sdram_ack,
                                o_ram_ce_n, o_ram_oe_n, o_ram_we_n, o_ram_sel,
                                o_ram_addr[4:0],
                                (!o_ram_oe_n) ? i_ram_data[15:0]
                                        : o_ram_data[15:0]
                                };
        wbscope #(.LGMEM(4), .SYNCHRONOUS(1), .HOLDOFFBITS(4))
                sdramscopei(i_clk, 1'b1, (!o_ram_ce_n), sdram_debug,
                        i_clk, wb_cyc, (wb_stb)&&(sdramscope_sel),
                        wb_we, wb_addr[0], wb_data,
                        sdramscope_ack, sdramscope_stall, sdramscope_data,
                        sdramscope_int);
```

This is what get added to design.h

18,20d17
< #define SDRAM_SCOPE
< #endif
< #ifdef SDRAM_ACCESS
45,47d41
< #ifdef SDRAM_SCOPE
< #else// SDRAM_SCOPE
< #endif // SDRAM_SCOPE

~~Trying to add sdram support to the catzip. Steps~~
~~The command "python main_sdramdev.py"~~
~~creates sdramdev.v & Sdcntl.v Verilog files and the testbench main_sdramdev_tb.vcd.~~
~~Steps to obtain the Values Change Dump VCD. This creates the files sdramdev.v & Sdcntl.v~~
~~Modified the /testbuilds/catzip_simulated/catzip/sim/verilated/Makefile to add the sdramsim.cpp to /testbuilds/catzip_simulated/catzip/sim/verilated/arm-main_tb.~~
~~The disassemble is found at~~
~~Appendix D: Disassemble of testbuilds/catzip_simulation/catzip/sim/verilated/arm-main_tb where the SDRAMSIM~~

~~***Note: Only one sdramdev_inst.convert(name = 'sdramdev') or SdCntl_inst.convert(name = 'Sdcntl')***~~
~~***can be uncommeted at the same time***~~
~~***not both or the following error will occur***~~
~~***Signal has multiple drivers: host_intf_rst_i***~~

~~In the directory /home/pi/testbuilds/learning_hdl/MyHDL/sdram~~
~~"rm -f *.vcd"~~
~~"python main_sdramdev.py"~~
~~"gtkwave main_sdramdev_tb.vcd"~~

Create sdramdev.txt this is currently a draft version for use with /home/pi/testbuilds/catzip_simulated/catzip/auto-data for the catzip.
/home/pi/testbuilds/catzip_simulated/catzip/auto-data for the catzip.
/home/pi/testbuilds/catzip_simulated/catzip/make autodata
Copying auto-data/toplevel.v to rtl/catzip/toplevel.v
Copying auto-data/main.v to rtl/catzip/main.v
Copying auto-data/regdefs.h to sw/host/regdefs.h
Copying auto-data/regdefs.cpp to sw/host/regdefs.cpp
Copying auto-data/board.h to sw/board/board.h
Copying auto-data/board.ld to sw/board/board.ld
Copying auto-data/rtl.make.inc to rtl/catzip/auto.mk
Copying auto-data/main_tb.cpp to sim/verilated/main_tb.cpp
Copying auto-data/testb.h to sim/verilated/testb.h
Steps to obtain the Values Change Dump VCD. This creates the file sdramdev.v.

Add the sdramsim.cpp & sdramsim.h to /testbuilds/catzip_simulated/catzip/sim/verilated.

Steps used to get a working catzip
"cd catzip"
". ./myenv.sh"
"make datestamp"

“make autodata”
“cd rtl/catzip”
“make” This command creates catzip.bin that can downloaded to FPGA. The Yosys Tools were used.
“cd ../../sim/verilated”
“make” This command creates arm-main_tb used to simulate the HX8K FPGA
“cd ../../sw/host” This creates
”make” This command creates arm-netpport & arm-wbregs

The script restores “config_catzip_simulation.sh” located in the learning_hdl/MyHDL/sdram folder. Starts with a working catzip

1st shell
. /home/pi/testbuilds/catzip_simulation/catzip/myenv.sh
/home/pi/testbuilds/catzip_simulation/catzip/
make autodata
2nd shell
. /home/pi/testbuilds/catzip_simulation/catzip/myenv.sh
/home/pi/testbuilds/catzip_simulation/catzip/rtl/catzip
make clean
make cpudefs.h
make design.h
make verilated
make bin
/home/pi/testbuilds/catzip_simulation/catzip/sim/verilated/
make clean
make
3rd shell
. /home/pi/testbuilds/catzip_simulation/catzip/myenv.sh
cd testbuilds/catzip_simulation/catzip/sim/verilated/
make clean
make
./arm-main_tb

The script used to execute the arm-wbregs command is found at Appendix A: Script that executes the arm-wbregs commands sim_hw_test.sh
The output of the arm-main_tb is found at Appendix B: CATZIP testing the verilator simulation.
Listening on port 8363
Listening on port 8364
> T

In 4th shell
. /home/pi/testbuilds/catzip_simulation/catzip/myenv.sh
cd testbuilds/catzip_simulation/catzip/sw/host/
./sim_hw_test.sh

The output of the sim_hw_test.sh is found at Appendix C: CATZIP testing the verilator simulation running wbregs command from the script
sim_hw_test.sh .

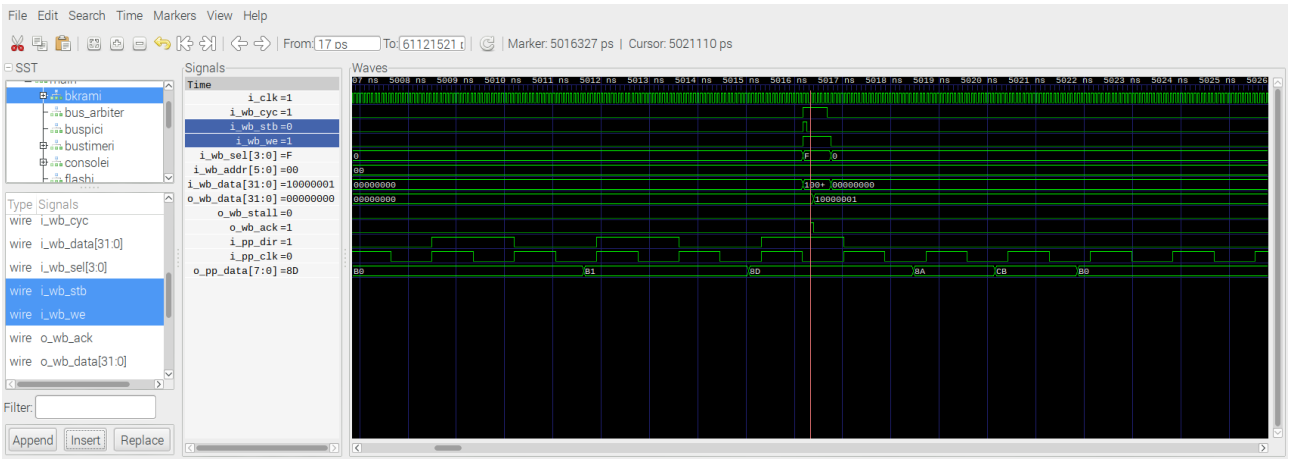
cp testbuilds/learning_hdl/MyHDL/sdram/ topsdcntl.v & SdramCntl.v ~/testbuilds/catzip_simulation/catzip/rtl/catzip/

cp ~/testbuilds/learning_hdl/MyHDL/sdram/sdramdev.txt ~/testbuilds/catzip_simulation/catzip/auto-data/

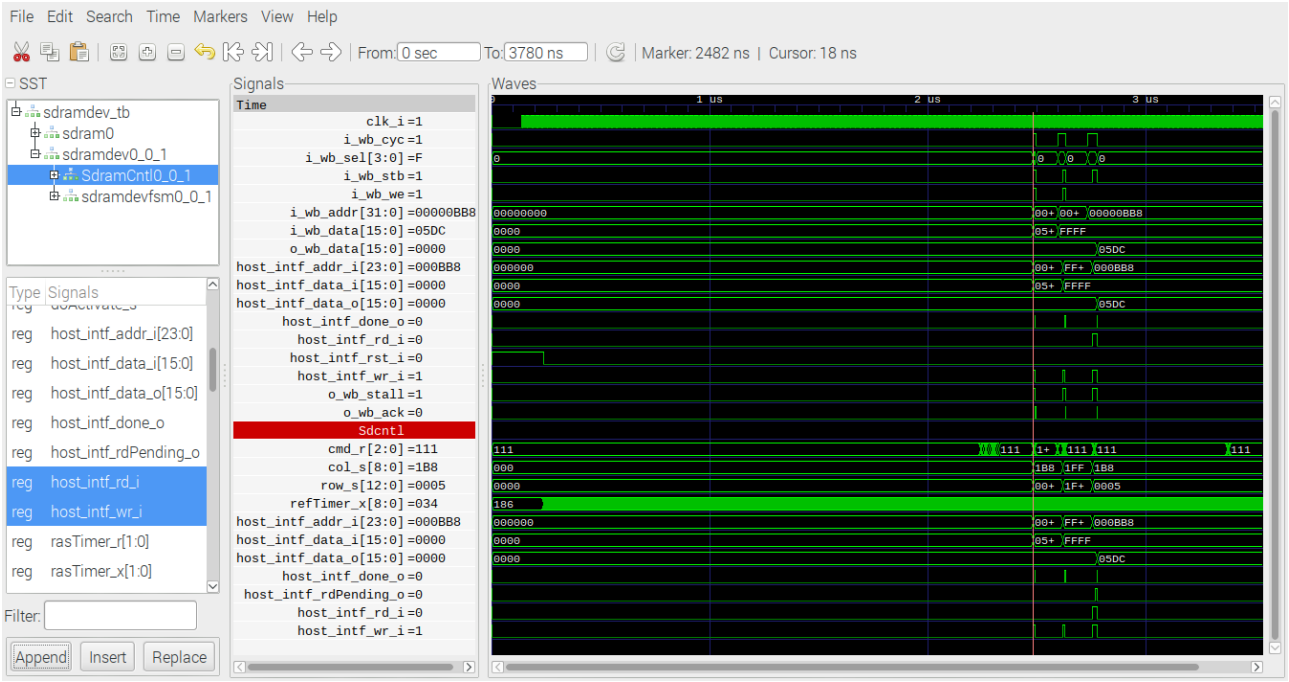
cp ~/testbuilds/learning_hdl/MyHDL/sdram/Makefile_autodata ~/testbuilds/catzip_simulation/catzip/auto-data/Makefile

Note This provides a verilator sim of both sram & blkram
. ~/testbuilds/catzip_simulation/catzip/myenv.sh
cd ~/testbuilds/catzip_simulation/catzip/autodata

Create sdcntl.txt
Modified Makefile in ~/testbuilds/catzip_simulation/catzip/autodata → Makefile_autodata



The Catboard has HX8K FPGA with a SDRAM that provides 0x000000 to 0xFFFFFF 16 bit locations.
The SDCONTROLLER takes care of refreshing the SDRAM and interface between the FPGA and the SDRAM. The data 0x05DC 1500 is being wrote to address 0x000BB8 3000, the data 0xFFFF 65535 is being wrote to address 0xFFFFFF 16777215. Then the data is read from address 0x000BB8 3000.



This test writes two SDRAM address and reads the value wrote to first location.
The data 0x05DC 1500 is being wrote to address 0x00000BB8 3000, 0xFFFF 65535 is being wrote to address 0x00FFFFFF 16777215. This is followed by reading the value at address 0x00000BB8 3000 which was 0x05DC 1500.

```
BANK 0 STATE : [CHANGE] Uninitialized -> Initialized @  2138
 BANK 1 STATE : [CHANGE] Uninitialized -> Initialized @  2138
 BANK 2 STATE : [CHANGE] Uninitialized -> Initialized @  2138
 BANK 3 STATE : [CHANGE] Uninitialized -> Initialized @  2138
-------------------------
 Mode   | CAS   |   Burst
--------|-------|---------
 Burst  | 3     |   1
-------------------------
 SDRAM : Bank 0 has active row 0005
 BANK 0 STATE : WRITING @  2514
 DATA : [WRITE] Addr: 440  Data: 1500
```

The data 0x05DC 1500 is being wrote to address 0x000BB8 3000 Row 0005 col Addr: 440 0x1B8 is where the 1500 0x05DC is written.

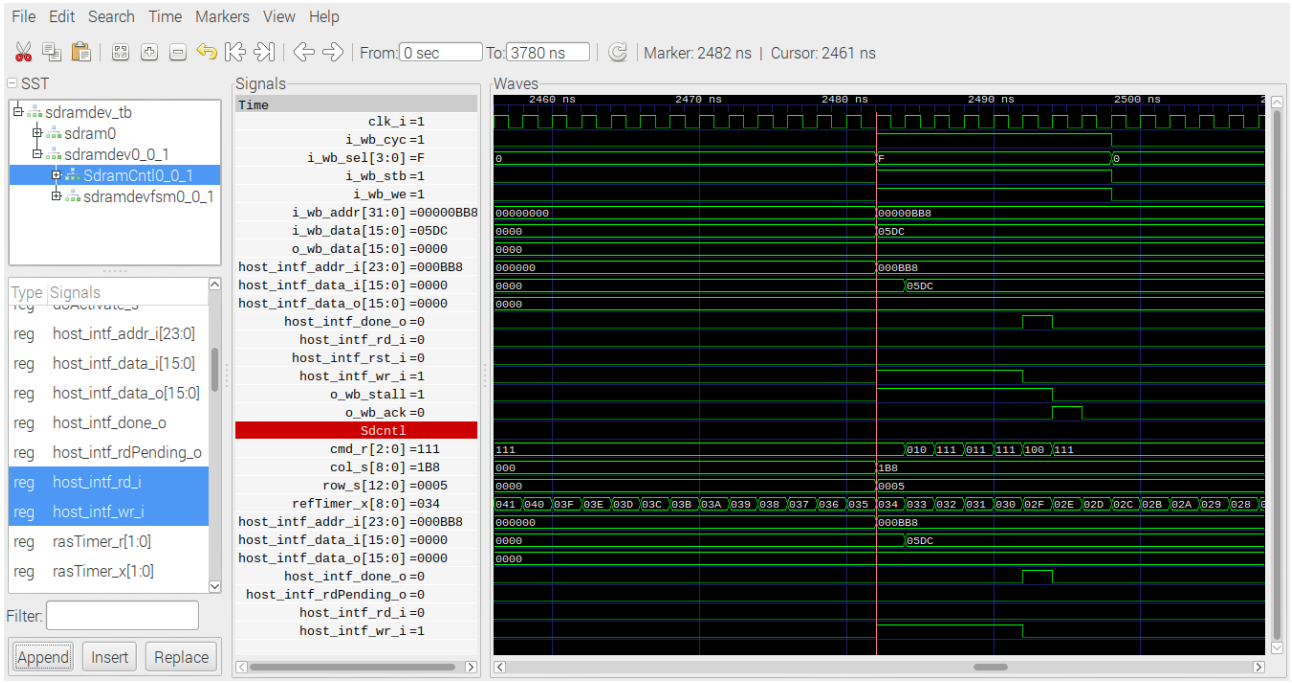
```
SDRAM : Bank 3 has active row 1fff
 BANK 3 STATE : WRITING @  2652
 DATA : [WRITE] Addr: 511  Data: 65535
 BANK 3 STATE : WRITING TO 01ff:511
```

The data 0xFFFF 65535 is being wrote to address 0xFFFFFF 16777215 (this is the last location in the sdram) Row 1FFF col Addr: 511 0x1FF is where the 65535 0xFFFF is written.

```
SDRAM : Bank 0 has active row 0005
 BANK 0 STATE : READING @  2790
 SDRAM : [READ] Commnad registered
 BANK 0 STATE : READING FROM 01b8:440
 STATE : [READ] Data Ready @  2794  value :  1500
```

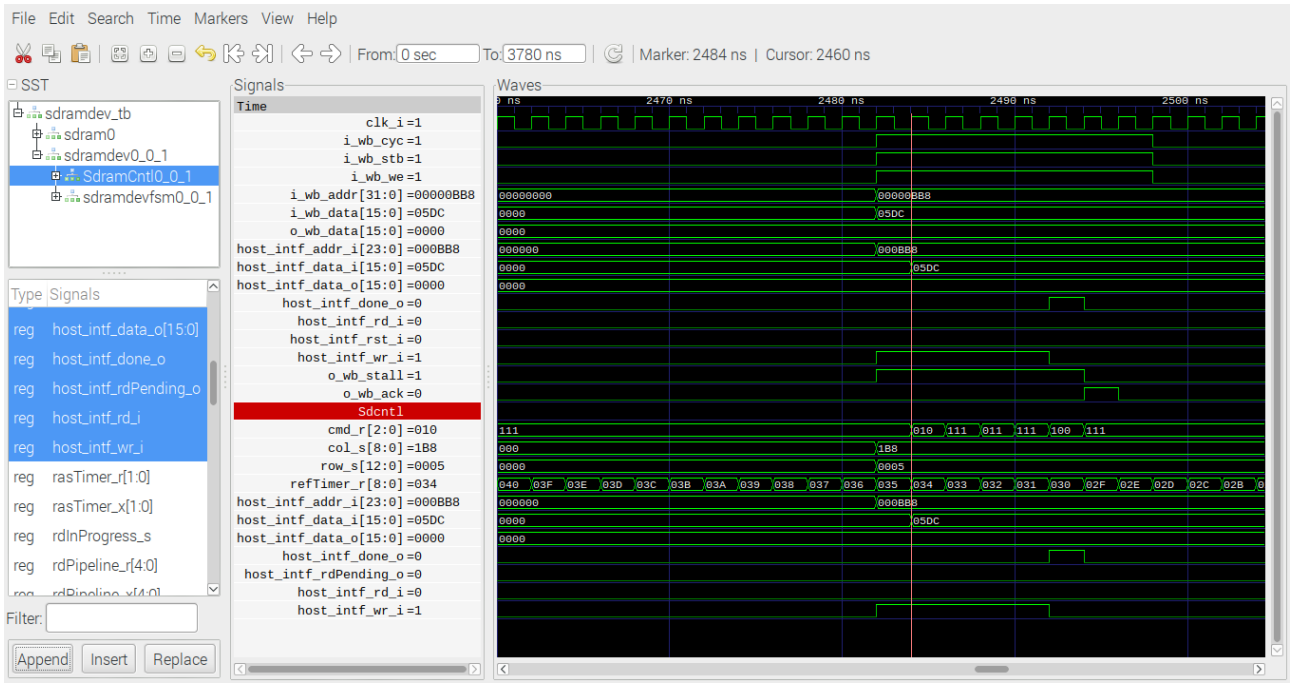
BANK 0 STATE : WRITING TO 01b8:440
4400440: 1500
SDRAM : Bank 3 has active row 1fff
BANK 3 STATE : WRITING @ 2652
DATA : [WRITE] Addr: 511 Data: 65535
BANK 3 STATE : WRITING TO 01ff:511
4400440: 1500
5110511: 65535
SDRAM : Bank 0 has active row 0005
BANK 0 STATE : READING @ 2790
SDRAM : [READ] Commnad registered
BANK 0 STATE : READING FROM 01b8:440
STATE : [READ] Data Ready @ 2794 value : 1500

In the image below the data 0x05DC 1500 is being wrote to address 0x00000BB8 3000, 0xFFFF 65535 is being wrote to address 0x00FFFFFF 16777215. This is followed by reading the value at address 0x00000BB8 3000 which was 0x05DC 1500.
BANK 3 STATE : WRITING @ 2652

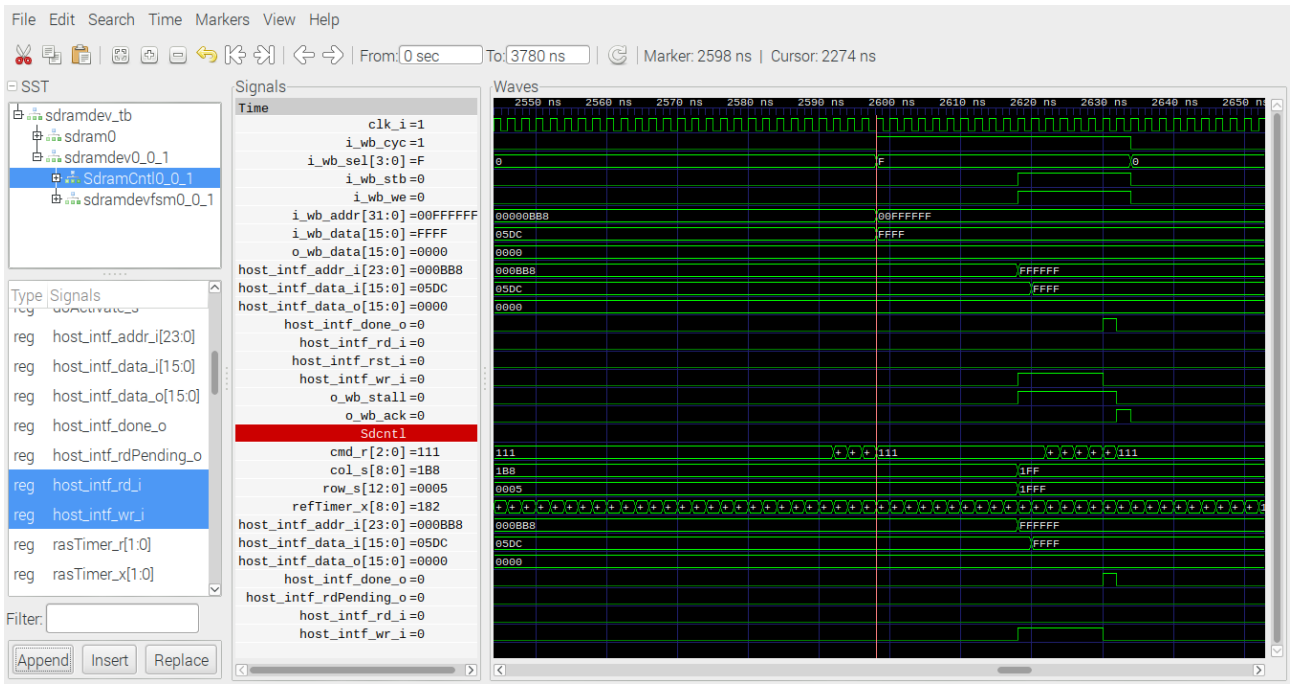


DATA : [WRITE] Addr: 511 Data: 65535
BANK 3 STATE : WRITING TO 01ff:511

SDRAM : Bank 0 has active row 0005
BANK 0 STATE : WRITING @ 2514
DATA : [WRITE] Addr: 440 Data: 1500
BANK 0 STATE : WRITING TO 01b8:440
In the image below the data 0x05DC 1500 is being wrote to address 0x000BB8 3000

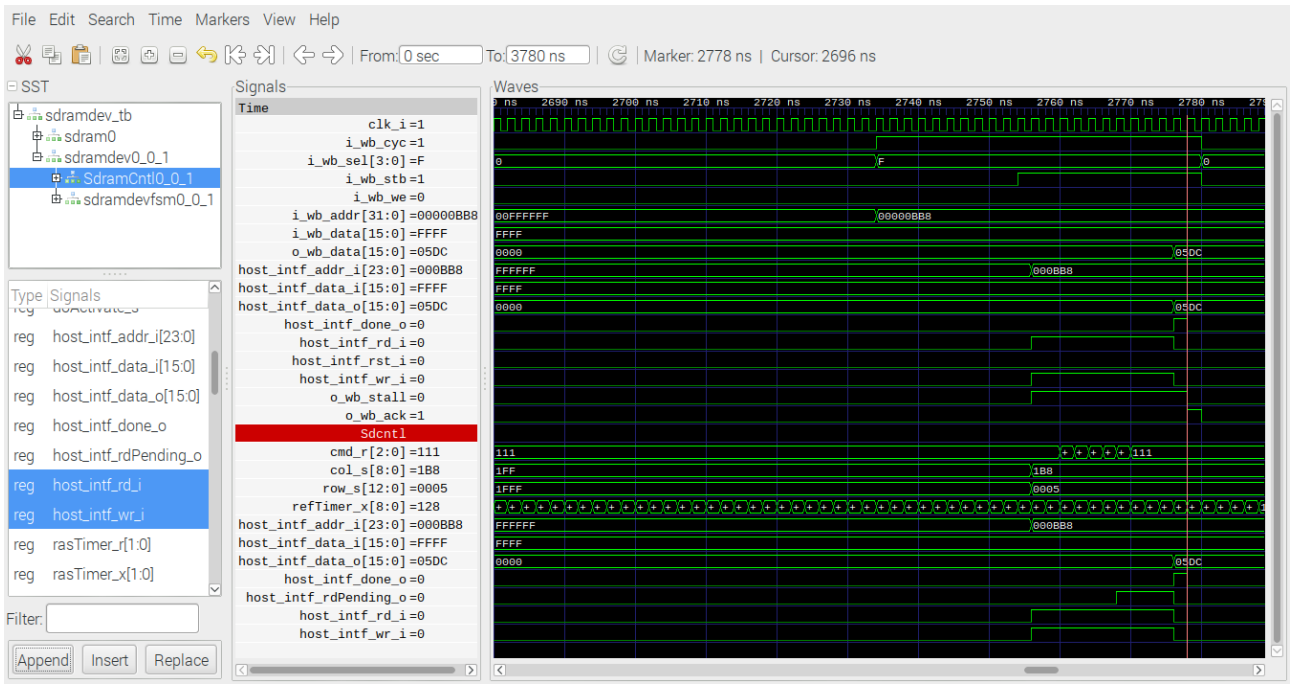


SDRAM : Bank 3 has active row 1fff
 BANK 3 STATE : WRITING @ 2652
 DATA : [WRITE] Addr: 511 Data: 65535
 BANK 3 STATE : WRITING TO 01ff:511
In the image below the data 0xFFFF 65535 is being wrote to address 0xFFFFFF 16777215.



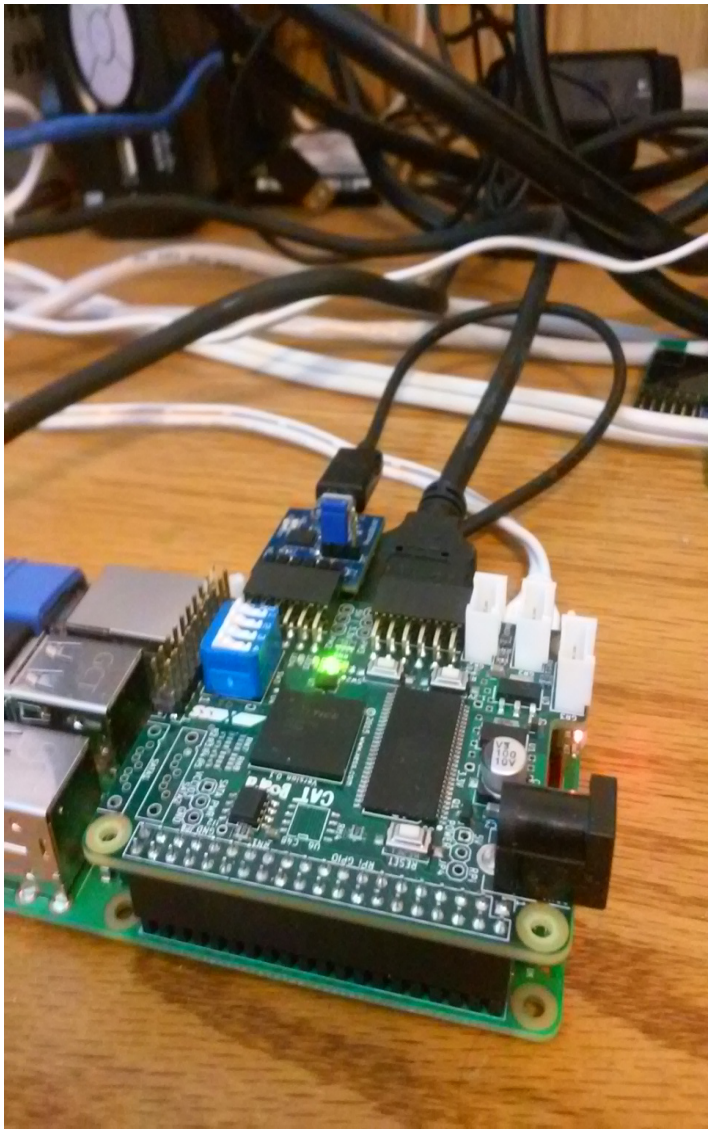
BANK 0 STATE : READING @ 2790
 SDRAM : [READ] Commnad registered
 BANK 0 STATE : READING FROM 01b8:440
 STATE : [READ] Data Ready @ 2794 value : 1500

In the image below reading the value at address 0x00000BB8 3000 which was 0x05DC 1500.



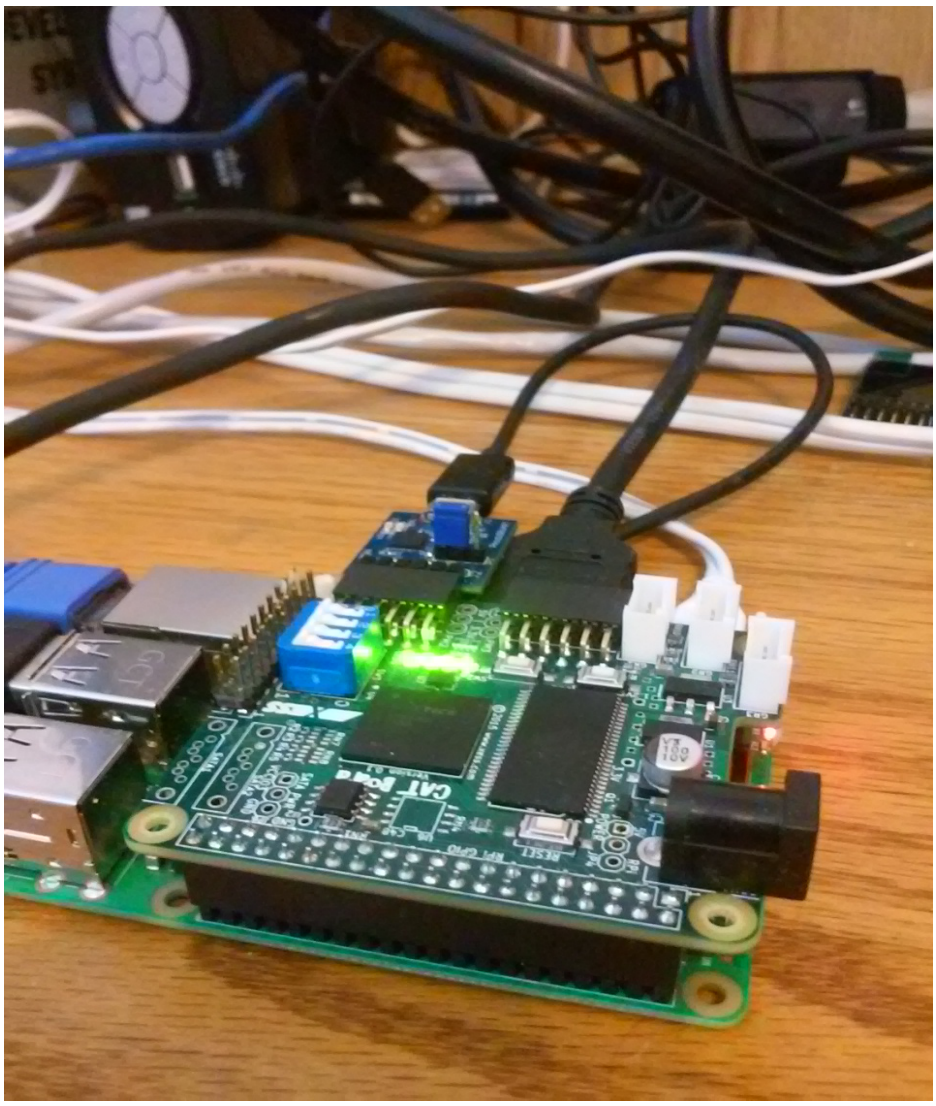
The memory_test takes approximately 3.5 sec or 175e6 clocks to complete at 50MHz. The memory_test writes to 16777215 locations and reads the data written.

Catboard running top.bin after the program has been load starting to write to memory during the memory_test



Catboard running top.bin after the program has been load starting to read from memory during the memory_test

Catboard running top.bin after the program has been load and the end memory memory_test. Note all 4 leds are on.



Now the sdram_test is memdev

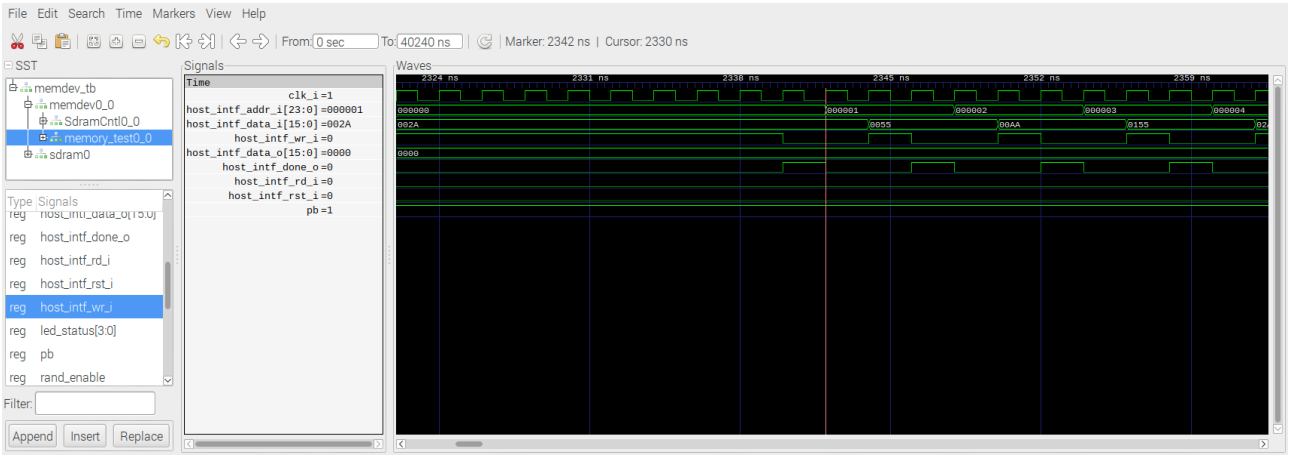
```
module top (
    clk100MHz,
    sdram_clk,
    sdram_return_clk,
    led_status,
    pb,
    memdev0_SdramCntl0_sd_intf_cke,
    memdev0_SdramCntl0_sd_intf_we,
    memdev0_SdramCntl0_sd_intf_addr,
    memdev0_SdramCntl0_sd_intf_dqml,
    memdev0_SdramCntl0_sd_intf_cas,
    memdev0_SdramCntl0_sd_intf_dqmh,
    memdev0_SdramCntl0_sd_intf_ras,
    memdev0_SdramCntl0_sd_intf_bs,
    memdev0_SdramCntl0_sd_intf_cs,
    memdev0_SdramCntl0_sd_intf_dq
);
=== top ===

   Number of wires:                441
```

```
   Number of wire bits:            968
   Number of public wires:          84
   Number of public wire bits:     537
   Number of memories:               0
   Number of memory bits:            0
   Number of processes:              0
   Number of cells:                719
     $_TBUF_                       16
     SB_CARRY                      68
     SB_DFF                         4
     SB_DFFE                        4
     SB_DFFER                      98
     SB_DFFES                       1
     SB_DFFESR                     45
     SB_DFFESS                      4
     SB_DFFR                       52
     SB_DFFS                        8
     SB_DFFSR                       3
     SB_DFFSS                       4
     SB_LUT4                      412
```

Host interface signals
wire [23:0] memdev0_SdramCntl0_host_intf_addr_i;
wire [15:0] memdev0_SdramCntl0_host_intf_data_i;
wire [15:0] memdev0_SdramCntl0_host_intf_data_i;
wire [15:0] memdev0_SdramCntl0_host_intf_data_o;
wire memdev0_SdramCntl0_host_intf_done_o;
wire memdev0_SdramCntl0_host_intf_wr_i;
wire memdev0_memory_test0_host_intf_rst_i;

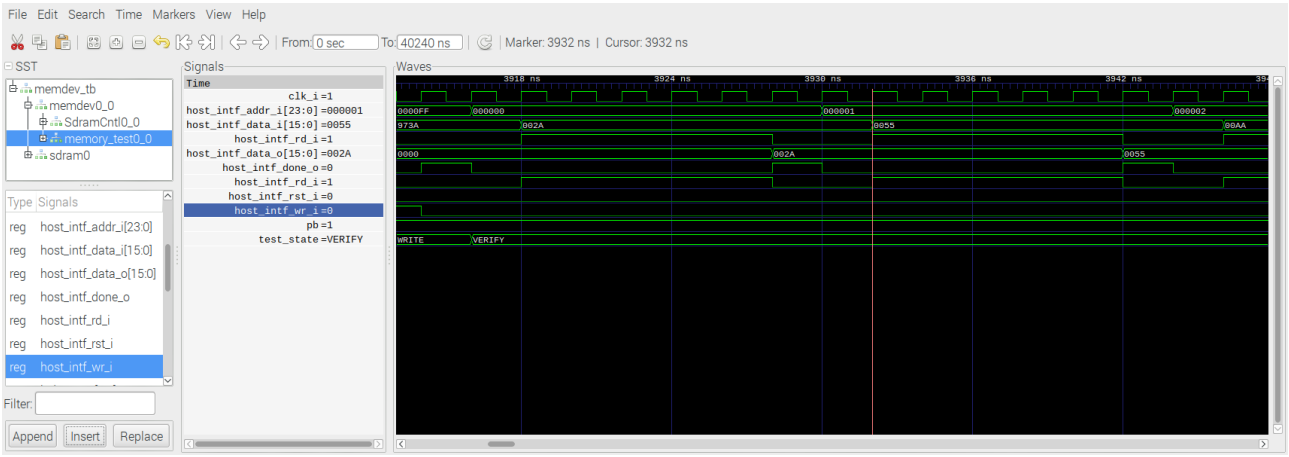
Write function
during a write
4 clks in between address chg
4 clks in between data chg
80 nsec

The 24 bit memdev0_SdramCntl0_host_intf_addr_i is set on the raising edge of clk. One clk later the 16 bit
memdev0_SdramCntl0_host_intf_data_i and memdev0_SdramCntl0_host_intf_wr_i goes hi. A clk later the memdev0_SdramCntl0_host_intf_done_o goes hi and the memdev0_SdramCntl0_host_intf_wr_i goes lo.



Read function

The 24 bit memdev0_SdramCntl0_host_intf_addr_i is set on the raising edge of clk. One clk later the 16 bit memdev0_SdramCntl0_host_intf_rd_i goes hi. Five clk later the memdev0_SdramCntl0_host_intf_done_o goes hi, memdev0_SdramCntl0_host_intf_data_o is valid and the memdev0_SdramCntl0_host_intf_rd_i goes lo.



During the first part of the memory_test Write takes place.

python memdev.py

make
sudo config_cat top.bin

2^24 16777216 0x000000 0xFFFFFF 16Bits
@PREFIX=sdram
@DEVID=SDRAM
@$LGMEMSZ=24
@LGMEMSZ.FORMAT=%d
@$NADDR=(1<<(@$THIS.LGMEMSZ-2))
@$NBYTES=(1<<(@$THIS.LGMEMSZ))
@NBYTES.FORMAT=0x%08x
@ACCESS=@$(DEVID)_ACCESS

```
@SLAVE.TYPE=MEMORY
@SLAVE.BUS=wb
@LD.PERM=wx
#!/bin/bash
```

Appendix A: Script that executes the arm-wbregs commands sim_hw_test.sh

```
#!/bin/bash


echo "The date built"
./arm-wbregs version
sleep 2
./arm-wbregs 0x700 0x10000001
sleep 2
./arm-wbregs 0x704 0x10000002
sleep 2
./arm-wbregs 0x708 0x10000003
sleep 2
./arm-wbregs 0x70c 0x10000004
sleep 2
./arm-wbregs 0x700
sleep 2
./arm-wbregs 0x704
sleep 2
./arm-wbregs 0x708
sleep 2
./arm-wbregs 0x70c
sleep 2
echo "Turning on the 4th led "
./arm-wbregs gpio 0x00010001
sleep 2
echo "Turning on the 1st led "
./arm-wbregs gpio 0x00020002
sleep 2
echo "Turning on the 2nd led "
./arm-wbregs gpio 0x00040004
sleep 5
echo "Turning off the leds "
./arm-wbregs gpio 0x00070000
```

Appendix B: CATZIP testing the verilator simulation.

```
./arm-main_tb
Listening on port 8363
Listening on port 8364
Accepted CMD connection
POLL = 1
RCVD: 6 bytes
< A611R
> A00000611R20180812
POLL = 1
RCVD: 0 bytes
```

< [CLOSED]
Accepted CMD connection
POLL = 1
RCVD: 14 bytes
< A701W10000001
> A00000701
> K00000000
POLL = 1
RCVD: 0 bytes
< [CLOSED]
Accepted CMD connection
POLL = 1
RCVD: 14 bytes
< A705W10000002
> A00000705
> K00000000
POLL = 1
RCVD: 0 bytes
< [CLOSED]
Accepted CMD connection
POLL = 1
RCVD: 14 bytes
< A709W10000003
> A00000709
> K00000000
POLL = 1
RCVD: 0 bytes
< [CLOSED]
Accepted CMD connection
POLL = 1
RCVD: 14 bytes
< A70dW10000004
> A0000070d
> K00000000
POLL = 1
RCVD: 0 bytes
< [CLOSED]
Accepted CMD connection
POLL = 1
RCVD: 6 bytes
< A701R
> A00000701R10000001
POLL = 1
RCVD: 0 bytes
< [CLOSED]
Accepted CMD connection
POLL = 1
RCVD: 6 bytes
< A705R
> A00000705R10000002
POLL = 1
RCVD: 0 bytes

< [CLOSED]
Accepted CMD connection
POLL = 1
RCVD: 6 bytes
< A709R
> A00000709R10000003
POLL = 1
RCVD: 0 bytes
< [CLOSED]
Accepted CMD connection
POLL = 1
RCVD: 6 bytes
< A70dR
> A0000070dR10000004
POLL = 1
RCVD: 0 bytes
< [CLOSED]
Accepted CMD connection
POLL = 1
RCVD: 11 bytes
< A609W10001
> A00000609K00000000
POLL = 1
RCVD: 0 bytes
< [CLOSED]
Accepted CMD connection
POLL = 1
RCVD: 11 bytes
< A609W20002
> A00000609K00000000
POLL = 1
RCVD: 0 bytes
< [CLOSED]
Accepted CMD connection
POLL = 1
RCVD: 11 bytes
< A609W40004
> A00000609K00000000
POLL = 1
RCVD: 0 bytes
< [CLOSED]
Accepted CMD connection
POLL = 1
RCVD: 11 bytes
< A609W70000
> A00000609K00000000
POLL = 1
RCVD: 0 bytes
< [CLOSED]
> Z
CMD: Only sent 0 bytes of 3
 ^C

Appendix C: CATZIP testing the verilator simulation running wbregs command from the script sim_hw_test.sh .

```
The date built
00000610 ( VERSION) : [....] 20180812
00000700 (     RAM)-> 10000001
00000704 (        )-> 10000002
00000708 (        )-> 10000003
0000070c (        )-> 10000004
00000700 (     RAM) : [....] 10000001
00000704 (        ) : [....] 10000002
00000708 (        ) : [....] 10000003
0000070c (        ) : [....] 10000004
Turning on the 4th led
00000608 (    GPIO)-> 00010001
Turning on the 1st led
00000608 (    GPIO)-> 00020002
Turning on the 2nd led
00000608 (    GPIO)-> 00040004
Turning off the leds
00000608 (    GPIO)-> 00070000
Chgs after adding sdcntl.txt & sdramdev.txt  blkram move from 0x700 to 0x00800000
The date built
00600010 ( VERSION) : [....] 20180812
00800000 (     RAM)-> 10000001
00800004 (        )-> 10000002
00800008 (        )-> 10000003
0080000c (        )-> 10000004
00800000 (     RAM) : [....] 10000001
00800004 (        ) : [....] 10000002
00800008 (        ) : [....] 10000003
0080000c (        ) : [....] 10000004
Turning on the 4th led
00600008 (    GPIO)-> 00010001
Turning on the 1st led
00600008 (    GPIO)-> 00020002
Turning on the 2nd led
00600008 (    GPIO)-> 00040004
Turning off the leds
00600008 (    GPIO)-> 00070000
```

Appendix D: Disassemble of testbuilds/catzip_simulation/catzip/sim/verilated/arm-main_tb where the SDRAMSIM

```
 00024654 <_ZN8SDRAMSIMclEiiiiiiijiss>:
   2466c:       1a000004        bne     24684 <_ZN8SDRAMSIMclEiiiiiiijiss+0x30>
   24678:       1a000002        bne     24688 <_ZN8SDRAMSIMclEiiiiiiijiss+0x34>
   24680:       ea00001e        b       24700 <_ZN8SDRAMSIMclEiiiiiiijiss+0xac>
   24698:       0a00001a        beq     24708 <_ZN8SDRAMSIMclEiiiiiiijiss+0xb4>
   246a4:       ca0000c3        bgt     249b8 <_ZN8SDRAMSIMclEiiiiiiijiss+0x364>
   246ac:       1a000034        bne     24784 <_ZN8SDRAMSIMclEiiiiiiijiss+0x130>
   246c4:       1a000038        bne     247ac <_ZN8SDRAMSIMclEiiiiiiijiss+0x158>
   246cc:       0a000031        beq     24798 <_ZN8SDRAMSIMclEiiiiiiijiss+0x144>
   246d8:       0a00002e        beq     24798 <_ZN8SDRAMSIMclEiiiiiiijiss+0x144>
```

```
  246e4:	0a00002b 	beq	24798 <_ZN8SDRAMSIMclEiiiiiiijiss+0x144>
  24708:	e59f496c 	ldr	r4, [pc, #2412]	; 2507c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa28>
  24718:	e59f0960 	ldr	r0, [pc, #2400]	; 25080
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa2c>
  24724:	e59f1958 	ldr	r1, [pc, #2392]	; 25084
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa30>
  24734:	e59f194c 	ldr	r1, [pc, #2380]	; 25088
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa34>
  24744:	e59f1940 	ldr	r1, [pc, #2368]	; 2508c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa38>
  24754:	e59f1934 	ldr	r1, [pc, #2356]	; 25090
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa3c>
  24764:	e59f1928 	ldr	r1, [pc, #2344]	; 25094
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa40>
  24770:	e59f3920 	ldr	r3, [pc, #2336]	; 25098
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa44>
  24778:	e59f191c 	ldr	r1, [pc, #2332]	; 2509c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa48>
  2477c:	e59f091c 	ldr	r0, [pc, #2332]	; 250a0
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa4c>
  24784:	e59f390c 	ldr	r3, [pc, #2316]	; 25098
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa44>
  2478c:	e59f1908 	ldr	r1, [pc, #2312]	; 2509c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa48>
  24790:	e59f090c 	ldr	r0, [pc, #2316]	; 250a4
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa50>
  24798:	e59f38f8 	ldr	r3, [pc, #2296]	; 25098
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa44>
  247a0:	e59f18f4 	ldr	r1, [pc, #2292]	; 2509c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa48>
  247a4:	e59f08fc 	ldr	r0, [pc, #2300]	; 250a8
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa54>
  247b0:	0a000019 	beq	2481c <_ZN8SDRAMSIMclEiiiiiiijiss+0x1c8>
  247b8:	0a000028 	beq	24860 <_ZN8SDRAMSIMclEiiiiiiijiss+0x20c>
  247c0:	0a000042 	beq	248d0 <_ZN8SDRAMSIMclEiiiiiiijiss+0x27c>
  247c8:	0a000057 	beq	2492c <_ZN8SDRAMSIMclEiiiiiiijiss+0x2d8>
  247d0:	1affffc8 	bne	246f8 <_ZN8SDRAMSIMclEiiiiiiijiss+0xa4>
  247d8:	1a000071 	bne	249a4 <_ZN8SDRAMSIMclEiiiiiiijiss+0x350>
  247e0:	1a00006f 	bne	249a4 <_ZN8SDRAMSIMclEiiiiiiijiss+0x350>
  247ec:	1a00006c 	bne	249a4 <_ZN8SDRAMSIMclEiiiiiiijiss+0x350>
  247f8:	0a000069 	beq	249a4 <_ZN8SDRAMSIMclEiiiiiiijiss+0x350>
  24804:	1afffbb 	bne	246f8 <_ZN8SDRAMSIMclEiiiiiiijiss+0xa4>
  24818:	ea00005d 	b	24994 <_ZN8SDRAMSIMclEiiiiiiijiss+0x340>
  24820:	1afffb4 	bne	246f8 <_ZN8SDRAMSIMclEiiiiiiijiss+0xa4>
  24828:	1afffb2 	bne	246f8 <_ZN8SDRAMSIMclEiiiiiiijiss+0xa4>
  24834:	0affffaf 	beq	246f8 <_ZN8SDRAMSIMclEiiiiiiijiss+0xa4>
  24840:	1affffac 	bne	246f8 <_ZN8SDRAMSIMclEiiiiiiijiss+0xa4>
  2485c:	eaffffa5 	b	246f8 <_ZN8SDRAMSIMclEiiiiiiijiss+0xa4>
  24868:	1a000009 	bne	24894 <_ZN8SDRAMSIMclEiiiiiiijiss+0x240>
  2487c:	aaffff9d 	bge	246f8 <_ZN8SDRAMSIMclEiiiiiiijiss+0xa4>
```

```
  24884:	e59f1820 	ldr	r1, [pc, #2080]	; 250ac
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa58>
  24890:	eafffff7 	b	24874 <_ZN8SDRAMSIMclEiiiiiiijiss+0x220>
  24898:	1a000007 	bne	248bc <_ZN8SDRAMSIMclEiiiiiiijiss+0x268>
  248a0:	1a000005 	bne	248bc <_ZN8SDRAMSIMclEiiiiiiijiss+0x268>
  248ac:	1a000002 	bne	248bc <_ZN8SDRAMSIMclEiiiiiiijiss+0x268>
  248b8:	1affff8e 	bne	246f8 <_ZN8SDRAMSIMclEiiiiiiijiss+0xa4>
  248bc:	e59f37d4 	ldr	r3, [pc, #2004]	; 25098
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa44>
  248c4:	e59f17d0 	ldr	r1, [pc, #2000]	; 2509c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa48>
  248c8:	e59f07e0 	ldr	r0, [pc, #2016]	; 250b0
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa5c>
  248d4:	1affff87 	bne	246f8 <_ZN8SDRAMSIMclEiiiiiiijiss+0xa4>
  248dc:	1affff85 	bne	246f8 <_ZN8SDRAMSIMclEiiiiiiijiss+0xa4>
  248e8:	1affff82 	bne	246f8 <_ZN8SDRAMSIMclEiiiiiiijiss+0xa4>
  248f4:	1affff7f 	bne	246f8 <_ZN8SDRAMSIMclEiiiiiiijiss+0xa4>
  24900:	1a000004 	bne	24918 <_ZN8SDRAMSIMclEiiiiiiijiss+0x2c4>
  24914:	eaffff77 	b	246f8 <_ZN8SDRAMSIMclEiiiiiiijiss+0xa4>
  24918:	e59f3778 	ldr	r3, [pc, #1912]	; 25098
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa44>
  24920:	e59f1774 	ldr	r1, [pc, #1908]	; 2509c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa48>
  24924:	e59f0788 	ldr	r0, [pc, #1928]	; 250b4
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa60>
  24930:	0a000007 	beq	24954 <_ZN8SDRAMSIMclEiiiiiiijiss+0x300>
  2493c:	1a000009 	bne	24968 <_ZN8SDRAMSIMclEiiiiiiijiss+0x314>
  24950:	eaffff68 	b	246f8 <_ZN8SDRAMSIMclEiiiiiiijiss+0xa4>
  24954:	e59f373c 	ldr	r3, [pc, #1852]	; 25098
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa44>
  2495c:	e59f1738 	ldr	r1, [pc, #1848]	; 2509c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa48>
  24960:	e59f0750 	ldr	r0, [pc, #1872]	; 250b8
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa64>
  2496c:	0affff61 	beq	246f8 <_ZN8SDRAMSIMclEiiiiiiijiss+0xa4>
  24970:	e59f3720 	ldr	r3, [pc, #1824]	; 25098
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa44>
  24978:	e59f171c 	ldr	r1, [pc, #1820]	; 2509c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa48>
  2497c:	e59f0738 	ldr	r0, [pc, #1848]	; 250bc
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa68>
  24988:	e59f171c 	ldr	r1, [pc, #1820]	; 250ac
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa58>
  2499c:	bafffff8 	blt	24984 <_ZN8SDRAMSIMclEiiiiiiijiss+0x330>
  249a0:	eaffff54 	b	246f8 <_ZN8SDRAMSIMclEiiiiiiijiss+0xa4>
  249a4:	e59f36ec 	ldr	r3, [pc, #1772]	; 25098
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa44>
  249ac:	e59f16e8 	ldr	r1, [pc, #1768]	; 2509c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa48>
  249b0:	e59f0708 	ldr	r0, [pc, #1800]	; 250c0
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa6c>
  249c4:	aa000005 	bge	249e0 <_ZN8SDRAMSIMclEiiiiiiijiss+0x38c>
```

```
  249dc:	eafffff6 	b	249bc <_ZN8SDRAMSIMclEiiiiiiijiss+0x368>
  249e4:	ea000000 	b	249ec <_ZN8SDRAMSIMclEiiiiiiijiss+0x398>
  249f0:	ca00001a 	bgt	24a60 <_ZN8SDRAMSIMclEiiiiiiijiss+0x40c>
  24a2c:	0affffed 	beq	249e8 <_ZN8SDRAMSIMclEiiiiiiijiss+0x394>
  24a48:	aaffffe6 	bge	249e8 <_ZN8SDRAMSIMclEiiiiiiijiss+0x394>
  24a4c:	e59f3644 	ldr	r3, [pc, #1604]	; 25098
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa44>
  24a54:	e59f1640 	ldr	r1, [pc, #1600]	; 2509c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa48>
  24a58:	e59f0664 	ldr	r0, [pc, #1636]	; 250c4
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa70>
  24a78:	0a000003 	beq	24a8c <_ZN8SDRAMSIMclEiiiiiiijiss+0x438>
  24a88:	0a000048 	beq	24bb0 <_ZN8SDRAMSIMclEiiiiiiijiss+0x55c>
  24a94:	da000010 	ble	24adc <_ZN8SDRAMSIMclEiiiiiiijiss+0x488>
  24aa0:	0a00000d 	beq	24adc <_ZN8SDRAMSIMclEiiiiiiijiss+0x488>
  24ac8:	1a000040 	bne	24bd0 <_ZN8SDRAMSIMclEiiiiiiijiss+0x57c>
  24af4:	1a000040 	bne	24bfc <_ZN8SDRAMSIMclEiiiiiiijiss+0x5a8>
  24b20:	1a000046 	bne	24c40 <_ZN8SDRAMSIMclEiiiiiiijiss+0x5ec>
  24b28:	1a000044 	bne	24c40 <_ZN8SDRAMSIMclEiiiiiiijiss+0x5ec>
  24b34:	1a000041 	bne	24c40 <_ZN8SDRAMSIMclEiiiiiiijiss+0x5ec>
  24b40:	0a00003e 	beq	24c40 <_ZN8SDRAMSIMclEiiiiiiijiss+0x5ec>
  24b4c:	e59f1558 	ldr	r1, [pc, #1368]	; 250ac
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa58>
  24b74:	1a000022 	bne	24c04 <_ZN8SDRAMSIMclEiiiiiiijiss+0x5b0>
  24b80:	1a000024 	bne	24c18 <_ZN8SDRAMSIMclEiiiiiiijiss+0x5c4>
  24b8c:	1a000026 	bne	24c2c <_ZN8SDRAMSIMclEiiiiiiijiss+0x5d8>
  24b98:	0afffed8 	beq	24700 <_ZN8SDRAMSIMclEiiiiiiijiss+0xac>
  24b9c:	e59f34f4 	ldr	r3, [pc, #1268]	; 25098
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa44>
  24ba4:	e59f14f0 	ldr	r1, [pc, #1264]	; 2509c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa48>
  24ba8:	e59f0518 	ldr	r0, [pc, #1304]	; 250c8
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa74>
  24bb0:	e59f34c4 	ldr	r3, [pc, #1220]	; 2507c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa28>
  24bc0:	e59f0504 	ldr	r0, [pc, #1284]	; 250cc
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa78>
  24bd4:	0affffbd 	beq	24ad0 <_ZN8SDRAMSIMclEiiiiiiijiss+0x47c>
  24be8:	0affffb8 	beq	24ad0 <_ZN8SDRAMSIMclEiiiiiiijiss+0x47c>
  24bf8:	eaffffb4 	b	24ad0 <_ZN8SDRAMSIMclEiiiiiiijiss+0x47c>
  24c00:	eaffffc0 	b	24b08 <_ZN8SDRAMSIMclEiiiiiiijiss+0x4b4>
  24c04:	e59f348c 	ldr	r3, [pc, #1164]	; 25098
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa44>
  24c0c:	e59f1488 	ldr	r1, [pc, #1160]	; 2509c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa48>
  24c10:	e59f04b8 	ldr	r0, [pc, #1208]	; 250d0
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa7c>
  24c18:	e59f3478 	ldr	r3, [pc, #1144]	; 25098
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa44>
  24c20:	e59f1474 	ldr	r1, [pc, #1140]	; 2509c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa48>
```

```
  24c24:	e59f04a8 	ldr	r0, [pc, #1192]	; 250d4
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa80>
  24c2c:	e59f3464 	ldr	r3, [pc, #1124]	; 25098
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa44>
  24c34:	e59f1460 	ldr	r1, [pc, #1120]	; 2509c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa48>
  24c38:	e59f0498 	ldr	r0, [pc, #1176]	; 250d8
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa84>
  24c44:	1a000025 	bne	24ce0 <_ZN8SDRAMSIMclEiiiiiiijiss+0x68c>
  24c4c:	1a000023 	bne	24ce0 <_ZN8SDRAMSIMclEiiiiiiijiss+0x68c>
  24c58:	0a000020 	beq	24ce0 <_ZN8SDRAMSIMclEiiiiiiijiss+0x68c>
  24c64:	1a00001d 	bne	24ce0 <_ZN8SDRAMSIMclEiiiiiiijiss+0x68c>
  24c70:	0a00000b 	beq	24ca4 <_ZN8SDRAMSIMclEiiiiiiijiss+0x650>
  24c7c:	cafffe9f 	bgt	24700 <_ZN8SDRAMSIMclEiiiiiiijiss+0xac>
  24ca0:	eaffffff3 	b	24c74 <_ZN8SDRAMSIMclEiiiiiiijiss+0x620>
  24cac:	1a000006 	bne	24ccc <_ZN8SDRAMSIMclEiiiiiiijiss+0x678>
  24cc8:	eafffe8c 	b	24700 <_ZN8SDRAMSIMclEiiiiiiijiss+0xac>
  24ccc:	e59f33c4 	ldr	r3, [pc, #964]  ; 25098
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa44>
  24cd4:	e59f13c0 	ldr	r1, [pc, #960]  ; 2509c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa48>
  24cd8:	e59f03fc 	ldr	r0, [pc, #1020]	; 250dc
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa88>
  24ce4:	1a000032 	bne	24db4 <_ZN8SDRAMSIMclEiiiiiiijiss+0x760>
  24cec:	1a000030 	bne	24db4 <_ZN8SDRAMSIMclEiiiiiiijiss+0x760>
  24cf8:	0a00002d 	beq	24db4 <_ZN8SDRAMSIMclEiiiiiiijiss+0x760>
  24d04:	0a00002a 	beq	24db4 <_ZN8SDRAMSIMclEiiiiiiijiss+0x760>
  24d10:	1a000016 	bne	24d70 <_ZN8SDRAMSIMclEiiiiiiijiss+0x71c>
  24d28:	1a000019 	bne	24d94 <_ZN8SDRAMSIMclEiiiiiiijiss+0x740>
  24d50:	e59f2388 	ldr	r2, [pc, #904]  ; 250e0
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa8c>
  24d6c:	eafffe63 	b	24700 <_ZN8SDRAMSIMclEiiiiiiijiss+0xac>
  24d78:	e59f32fc 	ldr	r3, [pc, #764]  ; 2507c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa28>
  24d88:	e59f0354 	ldr	r0, [pc, #852]  ; 250e4
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa90>
  24d90:	eaffffe5 	b	24d2c <_ZN8SDRAMSIMclEiiiiiiijiss+0x6d8>
  24d98:	e59f1348 	ldr	r1, [pc, #840]  ; 250e8
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa94>
  24d9c:	e59f02d8 	ldr	r0, [pc, #728]  ; 2507c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa28>
  24db0:	eafffffdd 	b	24d2c <_ZN8SDRAMSIMclEiiiiiiijiss+0x6d8>
  24db8:	1a00008a 	bne	24fe8 <_ZN8SDRAMSIMclEiiiiiiijiss+0x994>
  24dc0:	0a000088 	beq	24fe8 <_ZN8SDRAMSIMclEiiiiiiijiss+0x994>
  24dcc:	1a000085 	bne	24fe8 <_ZN8SDRAMSIMclEiiiiiiijiss+0x994>
  24dd8:	1a00003c 	bne	24ed0 <_ZN8SDRAMSIMclEiiiiiiijiss+0x87c>
  24de4:	1a00002a 	bne	24e94 <_ZN8SDRAMSIMclEiiiiiiijiss+0x840>
  24dfc:	0a000029 	beq	24ea8 <_ZN8SDRAMSIMclEiiiiiiijiss+0x854>
  24e30:	0a000021 	beq	24ebc <_ZN8SDRAMSIMclEiiiiiiijiss+0x868>
  24e60:	0afffe26 	beq	24700 <_ZN8SDRAMSIMclEiiiiiiijiss+0xac>
  24e88:	e59f3250 	ldr	r3, [pc, #592]  ; 250e0
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa8c>
```

```
  24e90:	eafffe1a 	b	24700 <_ZN8SDRAMSIMclEiiiiiiijiss+0xac>
  24e94:	e59f31fc 	ldr	r3, [pc, #508]  ; 25098
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa44>
  24e9c:	e59f11f8 	ldr	r1, [pc, #504]  ; 2509c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa48>
  24ea0:	e59f0234 	ldr	r0, [pc, #564]  ; 250dc
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa88>
  24ea8:	e59f31e8 	ldr	r3, [pc, #488]  ; 25098
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa44>
  24eb0:	e59f11e4 	ldr	r1, [pc, #484]  ; 2509c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa48>
  24eb4:	e59f0230 	ldr	r0, [pc, #560]  ; 250ec
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa98>
  24ebc:	e59f31d4 	ldr	r3, [pc, #468]  ; 25098
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa44>
  24ec4:	e59f11d0 	ldr	r1, [pc, #464]  ; 2509c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa48>
  24ec8:	e59f0220 	ldr	r0, [pc, #544]  ; 250f0 <_ZN8SDRAMSIMclEiiiiiiijiss+0xa9c>
  24ed8:	1a000033 	bne	24fac <_ZN8SDRAMSIMclEiiiiiiijiss+0x958>
  24ef0:	0a000032 	beq	24fc0 <_ZN8SDRAMSIMclEiiiiiiijiss+0x96c>
  24f04:	e59f21e8 	ldr	r2, [pc, #488]  ; 250f4 <_ZN8SDRAMSIMclEiiiiiiijiss+0xaa0>
  24f28:	1a000029 	bne	24fd4 <_ZN8SDRAMSIMclEiiiiiiijiss+0x980>
  24f7c:	0afffddf 	beq	24700 <_ZN8SDRAMSIMclEiiiiiiijiss+0xac>
  24fa0:	e59f3138 	ldr	r3, [pc, #312]  ; 250e0
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa8c>
  24fa8:	eafffdd4 	b	24700 <_ZN8SDRAMSIMclEiiiiiiijiss+0xac>
  24fac:	e59f30e4 	ldr	r3, [pc, #228]  ; 25098
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa44>
  24fb4:	e59f10e0 	ldr	r1, [pc, #224]  ; 2509c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa48>
  24fb8:	e59f011c 	ldr	r0, [pc, #284]  ; 250dc
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa88>
  24fc0:	e59f30d0 	ldr	r3, [pc, #208]  ; 25098
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa44>
  24fc8:	e59f10cc 	ldr	r1, [pc, #204]  ; 2509c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa48>
  24fcc:	e59f0118 	ldr	r0, [pc, #280]  ; 250ec
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa98>
  24fd4:	e59f30bc 	ldr	r3, [pc, #188]  ; 25098
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa44>
  24fdc:	e59f10b8 	ldr	r1, [pc, #184]  ; 2509c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa48>
  24fe0:	e59f0110 	ldr	r0, [pc, #272]  ; 250f8 <_ZN8SDRAMSIMclEiiiiiiijiss+0xaa4>
  24fec:	1afffdc3 	bne	24700 <_ZN8SDRAMSIMclEiiiiiiijiss+0xac>
  24ff4:	0a000005 	beq	25010 <_ZN8SDRAMSIMclEiiiiiiijiss+0x9bc>
  25000:	0a000002 	beq	25010 <_ZN8SDRAMSIMclEiiiiiiijiss+0x9bc>
  2500c:	1afffdbb 	bne	24700 <_ZN8SDRAMSIMclEiiiiiiijiss+0xac>
  25010:	e59f4064 	ldr	r4, [pc, #100]  ; 2507c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa28>
  25020:	e59f00d4 	ldr	r0, [pc, #212]  ; 250fc <_ZN8SDRAMSIMclEiiiiiiijiss+0xaa8>
  2502c:	e59f1054 	ldr	r1, [pc, #84]   ; 25088
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa34>
```

```
  2503c:	e59f1048 	ldr	r1, [pc, #72]	; 2508c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa38>
  2504c:	e59f103c 	ldr	r1, [pc, #60]	; 25090
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa3c>
  2505c:	e59f1030 	ldr	r1, [pc, #48]	; 25094
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa40>
  25068:	e59f3028 	ldr	r3, [pc, #40]	; 25098
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa44>
  25070:	e59f1024 	ldr	r1, [pc, #36]	; 2509c
<_ZN8SDRAMSIMclEiiiiiiijiss+0xa48>
  25074:	e59f0084 	ldr	r0, [pc, #132]	; 25100
<_ZN8SDRAMSIMclEiiiiiiijiss+0xaac>
```