ASIC ReRAM Test Chip
Documentation/Datasheet

Introduction

This document acts as a guide to using the ASIC ReRAM test chip designed by senior design team SDMay25-19 at Iowa State University. Information enclosed in this document includes C code documentation and pin definitions, as well as the pinout for the chip itself.

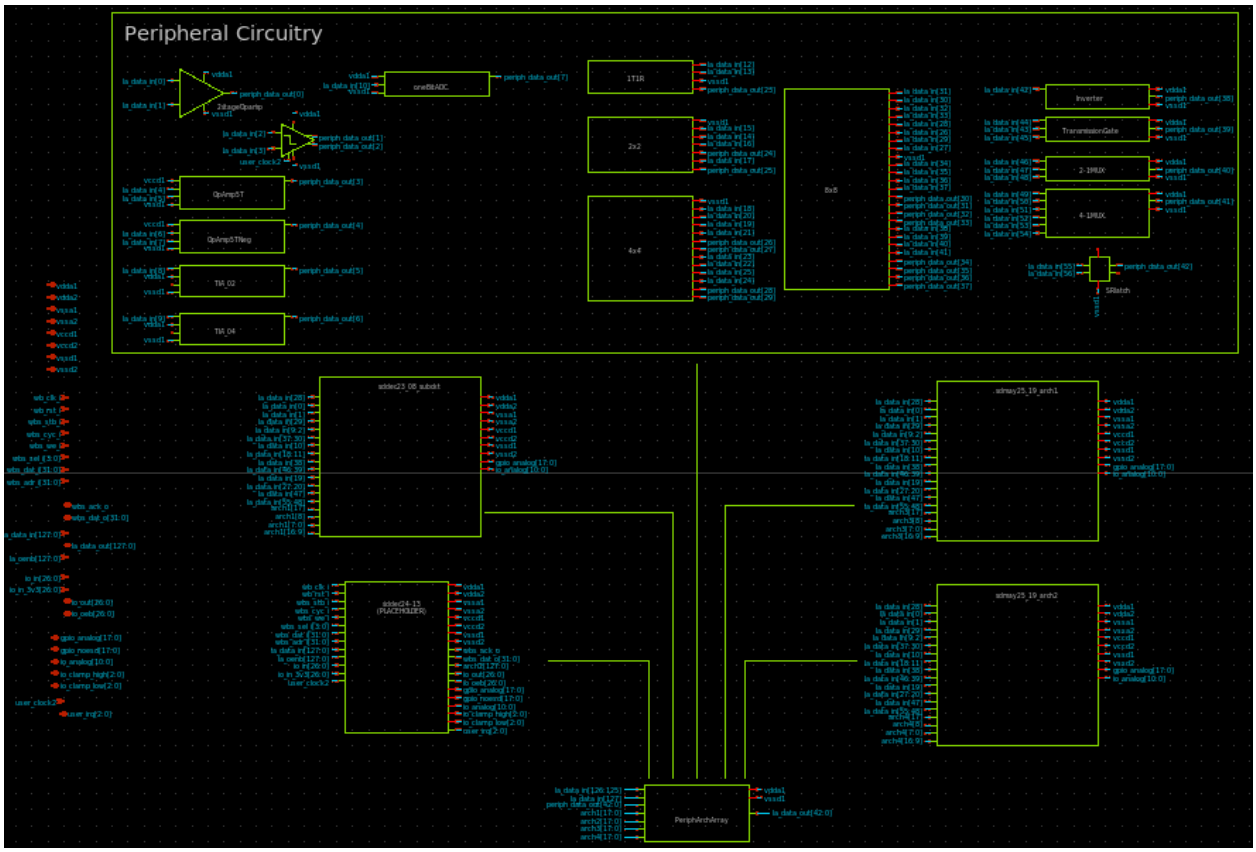

Figure 1: Top Level Schematic

The overall design of our chip consists of four ReRAM compute architectures, as well as a peripheral circuitry area with our component circuits individually mapped to pins. All components of this chip can be tested using the code and pin definitions provided in this document.

Code

Our repository contains two C files. The first is called “sdmay25_defs.h”, which contains helpful functions for writing and reading logic analyzer pins, as well as pin definitions for ease of use. The second file is an example “main.c”, which performs a few example functions on the chip to get the user started.

```
static inline void write_la_pin(uint32_t pin, uint32_t val)
```

*Use this function to write a HIGH or LOW **val**ue to a single **pin**.*

```
static inline void write_la_pins(uint32_t lsb_pin, uint32_t width,
uint32_t val)
```

*Use this function to write a **val**ue to the logic analyzer that spans more than a single bit. The pin definitions provide appropriate values for **lsb_pin** and **width** (see definitions table below).*

```
static inline uint32_t read_la_pin(uint32_t pin)
```

*Use this function to read a single logic analyzer **pin**.*

```
static inline uint32_t read_la_pins(uint32_t lsb_pin, uint32_t width)
```

*Use this function to read a value of a given **width** from the least significant pin (**lsb_pin**). The pin definitions provide appropriate values for **lsb_pin** and **width** (see definitions table below).*

See our repository for example main.c.

Definitions/Pinout

*Note: for signals wider than 1 bit, the full range of bits is noted next to the least significant bit pin. Width

sdmay_defs.h Definition	Caravel Pin	Description/Notes
PERIPH_SELECT	la_data_in[127]	1: peripheral output, 0: architecture output
PERIPH_DATA_OUT_LSB	la_data_out[42:0]	
PERIPH_DATA_OUT_WIDTH	N/A (value 43)	
TWO_STAGE_OP_AMP_IN_MINUS	la_data_in[0]	2 stage op amp Vin-
TWO_STAGE_OP_AMP_IN_PLUS	la_data_in[1]	2 stage op amp Vin+
TWO_STAGE_OP_AMP_OUT	la_data_out[0]	
COMPARATOR_IN_PLUS	la_data_in[2]	Comparator Vin+
COMPARATOR_IN_MINUS	la_data_in[3]	Comparator Vin-
COMPARATOR_OUT_PLUS	la_data_out[1]	Comparator Vout+
COMPARATOR_OUT_MINUS	la_data_out[2]	Comparator Vout-
OP_AMP_5T_IN_PLUS	la_data_in[4]	5T op amp Vin+
OP_AMP_5T_IN_MINUS	la_data_in[5]	5T op amp Vin-
OP_AMP_5T_NEG_IN_PLUS	la_data_in[6]	5T op amp neg Vin+
OP_AMP_5T_NEG_IN_MINUS	la_data_in[7]	5T op amp neg Vin-
OP_AMP_5T_OUT	la_data_out[3]	
OP_AMP_5T_NEG_OUT	la_data_out[4]	
TIA_02_IN	la_data_in[8]	
TIA_04_IN	la_data_in[9]	
TIA_02_OUT	la_data_out[5]	
TIA_04_OUT	la_data_out[6]	
ADC_IN	la_data_in[10]	

ADC_OUT	la_data_out[7]	
BL_IN_1T1R	la_data_in[12]	Bitline input
WL_IN_1T1R	la_data_in[13]	Worldline input
SL_OUT_1T1R	la_data_out[23]	Sourceline input
BL_IN_2X2_LSB	la_data_in[15:14]	
BL_IN_2X2_WIDTH	N/A (value 2)	
WL_IN_2X2_LSB	la_data_in[17:16]	
WL_IN_2X2_WIDTH	N/A (value 2)	
SL_OUT_2X2_LSB	la_data_out[25:24]	
SL_OUT_2X2_WIDTH	N/A (value 2)	
BL_IN_4X4_LSB	la_data_in[21:18]	
BL_IN_4X4_WIDTH	N/A (value 4)	
WL_IN_4X4_LSB	la_data_in[25:22]	
WL_IN_4X4_WIDTH	N/A (value 4)	
SL_OUT_4X4_LSB	la_data_out[29:26]	
SL_OUT_4X4_WIDTH	N/A (value 4)	
BL_IN_8X8_LSB	la_data_in[33:26]	
BL_IN_8X8_WIDTH	N/A (value 8)	
WL_IN_8X8_LSB	la_data_in[40:34]	
WL_IN_8X8_WIDTH	N/A (value 8)	
SL_OUT_8X8_LSB	la_data_out[37:30]	
SL_OUT_8x8_WIDTH	N/A (value 8)	
INVERTER_IN	la_data_in[42]	
INVERTER_OUT	la_data_out[38]	
TRANSMISSION_GATE_IN	la_data_in[43]	

TRANSMISSION_GATE_SP	la_data_in[44]	
TRANSMISSION_GATE_SN	la_data_in[45]	
TRANSMISSION_GATE_OUT	la_data_out[39]	
MUX_2_1_SEL	la_data_in[46]	0: output A, 1: output B
MUX_2_1_A	la_data_in[47]	
MUX_2_1_B	la_data_in[48]	
MUX_2_1_OUT	la_data_out[40]	
MUX_4_1_SEL1	la_data_in[49]	See next row
MUX_4_1_SEL2	la_data_in[50]	(With SEL1) 00: output A, 01: output B, 10: output C, 11: output D
MUX_4_1_A	la_data_in[51]	
MUX_4_1_B	la_data_in[52]	
MUX_4_1_C	la_data_in[53]	
MUX_4_1_D	la_data_in[54]	
MUX_4_1_OUT	la_data_out[41]	
SR_LATCH_S	la_data_in[55]	
SR_LATCH_R	la_data_in[56]	
SR_LATCH_Q	la_data_out[42]	
ARCH_SELECT_LSB	la_data_in[126:125]	00: sddec23-08. 01: sddec24-13, 10: sdmay25-19 arch1, 11: sdmay25-19 arch2
ARCH_SELECT_WIDTH	N/A (value 2)	
ARCH_DATA_OUT_LSB	la_data_out[17:0]	
ARCH_DATA_OUT_WIDTH	N/A (value 18)	
WRITE_SELECT_8LINEBITINPUT_1	la_data_in[0]	
WRITE_SELECT_8LINEBITINPUT_2	la_data_in[28]	

WRITE_FORM_SELECT_8LINEBITINPUT_1	la_data_in[1]	
WRITE_FORM_SELECT_8LINEBITINPUT_2	la_data_in[29]	
LA_IN_8LINEBITINPUT_1_LSB	la_data_in[9:2]	
LA_IN_8LINEBITINPUT_1_WIDTH	N/A (value 8)	
LA_IN_8LINEBITINPUT_2_LSB	la_data_in[37:30]	
LA_IN_8LINEBITINPUT_2_WIDTH	N/A (value 8)	
WRITE_SELECT_8LINEBITINPUT_1	la_data_in[10]	
WRITE_SELECT_8LINEBITINPUT_2	la_data_in[38]	
LA_IN_8LINEWORDINPUT_1_LSB	la_data_in[18:11]	
LA_IN_8LINEWORDINPUT_1_WIDTH	N/A (value 8)	
LA_IN_8LINEWORDINPUT_2_LSB	la_data_in[46:39]	
LA_IN_8LINEWORDINPUT_2_WIDTH	N/A (value 8)	
S_8LINESELECTINPUT_1	la_data_in[19]	
S_8LINESELECTINPUT_2	la_data_in[47]	
LA_IN_8LINESELECTINPUT_1_LSB	la_data_in[27:20]	
LA_IN_8LINESELECTINPUT_1_WIDTH	N/A (value 8)	
LA_IN_8LINESELECTINPUT_2_LSB	la_data_in[55:48]	
LA_IN_8LINESELECTINPUT_2_WIDTH	N/A (value 8)	
VSSNEG_8LINESELECTOUTPUT_02	la_data_in[8]	
VSSNEG_8LINESELECTOUTPUT_04	la_data_in[17]	
LA_OUT_8LINESELECTOUTPUT02_LSB	la_data_out[7:0]	
LA_OUT_8LINESELECTOUTPUT02_WIDTH	N/A (value 8)	
LA_OUT_8LINESELECTOUTPUT04_LSB	la_data_out[16:9]	
LA_OUT_8LINESELECTOUTPUT04_WIDTH	N/A (value 8)	