

## **LAB 2 DEMO – Working an SRAM, a Register File, and C Functions**

**Team** \_\_\_\_\_

**Reviewer** \_\_\_\_\_

### **Working with an SRAM (40)**

---

Driver – (Behavioural) Verilog model

Write – data 127..0 to addresses 0..127

Read– data 127..0 from addresses 0..127

Display on the RED LEDs

Signal Tap Interface – Verify

Write – data 127..0 to addresses 0..127

Read– data 127..0 to addresses 0..127

### **Working with a Register File (50)**

---

Design – (Structural) Verilog model – thirty two 32 bit registers

Stand Alone Test

Write hex values

0xFFFF000F...0xFFFF0000 to registers Register 0..Register 15

0x0000FFF0...0x0000FFFF to registers Register 16..Register 31

Read and display LS byte from...

KEY0 0: Register 0...Register 15 on the Red LEDs

KEY0 1: Register 16...Register 31 on the Red LEDs

## **Integrated SRAM and Register File (50)**

---

### **Integrated Register File and SRAM Test**

#### **Store to SRAM**

Write the binary data 0..127 to the first 128 locations in the SRAM – comprising 4 32 word blocks: 0..3.

Binary data 127 should go into address 0 and binary data 0 into address 127 in the SRAM.

#### **Transfer from Memory to Register File – Write to Single Register**

For SRAM block 0

Read words from the SRAM and write the 32 word block to the 32 registers.

#### **Read Two Registers Simultaneously**

For each block read and display LS byte from...

KEY0 0: Register 0..Register 15 on Red LEDS as Reg 1 Read Data

KEY0 1: Register 16..Register 31 on Red LEDS as Reg 2 Read Data

#### **Transfer from Register File to Memory – Read from Single Register**

For the designated blocks

Write Reg 1 Read Data values to SRAM 128..144 for B0 and 162..178 for B2

Write Reg 2 Read Data values to SRAM 145..161 for B1 and 179..195 for B3

#### **Signal Tap Interface**

Verify SRAM Write and Read using Signal Tap

---

## **C Language Functions (40)**

---

float getDuration(void)

float computeVelocity(float distance, float time)

float duration(float distance, float velocity, float headWind )

void displayResults(float duration)

Fully functional