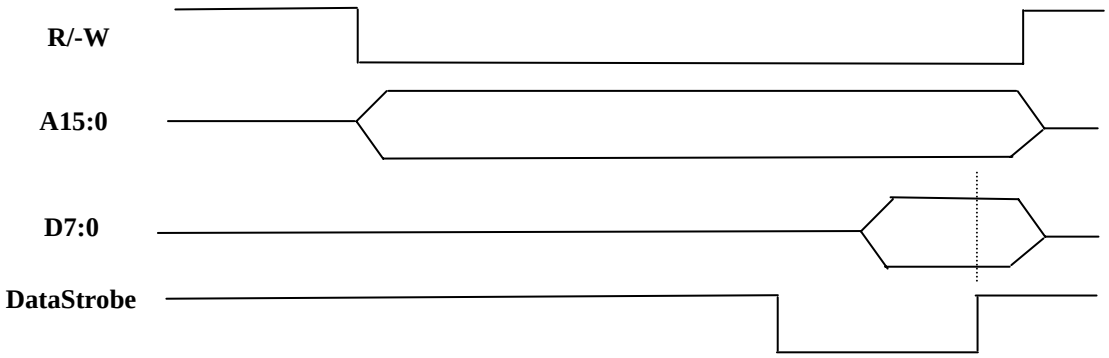
Ph.D Qualifying Exam (Microprocessor Applications) – January 2009

1. You are given a microprocessor with a 16-bit address bus (A15:0), an 8-bit data bus (D7:0), an active high chip enable (CE), an active high output enable (OE), a read/write input (1=read, 0=write), and an active low $\overline{\text{DataStrobe}}$ (where a rising edge represents valid data). **(a – 50%)** Using 4kx4 SRAM devices, show the devices along with the gate-level decode circuitry required to place 8k words (word = 8 bits) of memory at 4000 (hex) using full address decoding. **(b – 25%)** Specify the address range for the memory. A timing diagram for a write is given below to illustrate the $\overline{\text{DataStrobe}}$:



For a read, assume the microprocessor reads data on the rising edge of $\overline{\text{DataStrobe}}$. **(c – 25%)** In addition, create a byte-wide input port that is redundantly mapped (using partial address decoding) to addresses 2000 (hex) to 2fff (hex) using an octal tristate buffer. Assume the tristate buffer has an active high enable control pin.

2. Given a microprocessor with a 16-bit address and data bus, sixteen 16-bit registers (reg0, reg1, .. reg15), and the following instruction set architecture (ISA), write an assembly program that does the following. **(a – 25%)** Uses continuous polling to wait until a button mapped to bit 0 of port 2000 (hex) is pressed. Assume 0 corresponds to a pressed button and that debouncing is not needed. **(b – 25%)** After the button is pressed, the program should read a 16-bit sensor mapped to address 3000 (hex) every 250 ms. Assume that an interrupt service routine (ISR) is configured to execute every 250 ms after the button is pressed. **(c – 25%)** For every 4 sensor reads (i.e., once a second), the program should calculate the sum of the previous 4 sensor readings and write the sum to a 32 KB SRAM mapped to address 4000 (hex). Each sum written to the SRAM should be stored in a unique location (i.e., the first write should be to 4000, the second to 4001, etc.). **(d – 25%)** The program should terminate 30 seconds after the button is pressed. Provide code for the main application and for the ISR. Clearly define any memory locations used for variables. Do not use registers to communicate between the main application and the ISR. Clearly list any assumptions. Feel free to comment the code.

Microprocessor ISA:

16 bit Address Bus & 16-bit Data Bus
16 16-bit registers (reg0, reg1, ..., reg15)

Instruction	Operands	Description	Example
LD	regD, immediate(regA)	regD = memory[immediate + regA]	LD reg3, 1000(reg5)
LDI	regD, immediate	regD = immediate	LDI reg7, 0
ST	regD, immediate(regA)	memory[immediate + regA] = regD	ST reg8, 3000(reg2)
ADD	regD, regS1, regS2	regD = regS1 + regS2	ADD reg5, reg6, reg12
SUB	regD, regS1, regS2	regD = regS1 - regS2	SUB reg5, reg6, reg12
AND (bitwise)	regD, regS1, regS2	regD = regS1 and regS2	AND reg5, reg6, reg12
BEQ	regS1, regS2, label	Branch to label if regS1 = regS2	BEQ reg2, reg3, targetLabel
BLT	regS1, regS2, label	Branch to label if regS1 < regS2	BLT reg2, reg3, targetLabel
BLE	regS1, regS2, label	Branch to label if regS1 <= regS2	BLE reg2, reg3, targetLabel
BGT	regS1, regS2, label	Branch to label if regS1 > regS2	BGT reg2, reg3, targetLabel
BGE	regS1, regS2, label	Branch to label if regS1 >= regS2	BGE reg2, reg3, targetLabel
J	Label	Jump to label	J targetLabel