Tutorial One:
Master: SRAM Controller is the master
Slave: SRAM is the slave
*Nios 2 Soft Core could also be considered the master.

Tutorial Two:
Quartus---------- Hardware
Qsys---------- Hardware
Nios SBT for Eclipse---------- Software
SignalTap II---------- Both

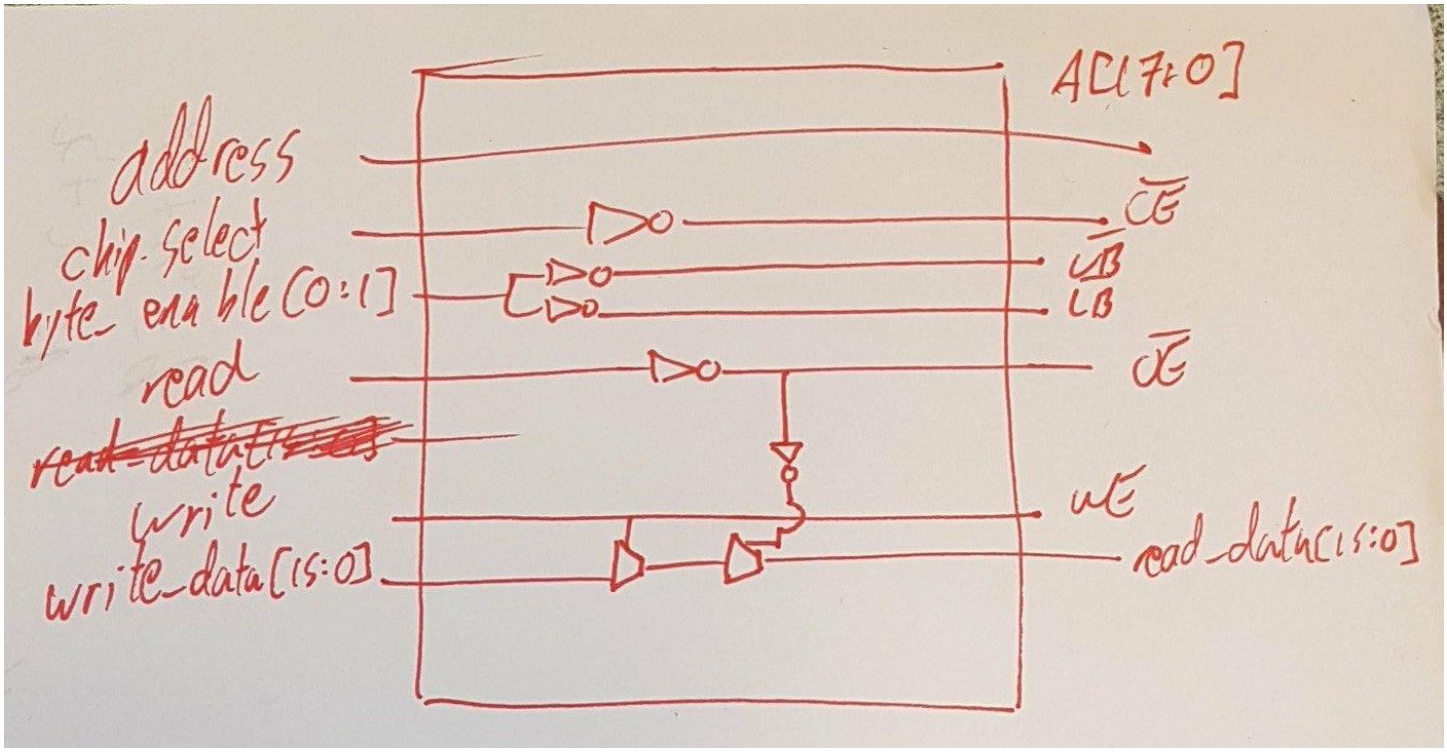
Question One:
512Kbytes = 512 000 bytes
The highest address is 512 000 - 1 = 511 999, or 0x7FFFF in hex.
The lowest address is simply 0.

If this was intended as 512 kilobits not 512 kilobytes than there are 64 kilobytes of memory.
If there are 64Kilobytes of memory then 65353 is the highest address in memory corresponding with 0xFFFF.

Question Two:
Lowest address is the base address at 0x0000

Question Three:
0x00000

Question Five:



Question 6:
module SRAM_Cont (

```
	//------Inputs------//
	input clk,
	input reset_n,
	input [17:0] address,
	input chipselect,
	input [1:0] byte_enable,
	input read,
	input write
	input [15:0] write_data,


	//-------Bidirectionals-------//
	inout [15:0] SRAM_DQ,

	//-------Output-------//
	output [15:0] read_data,
	output [17:0] SRAM_ADDR,
	output reg SRAM_CE_N,
	output reg SRAM_WE_N,
	output reg SRAM_OE_N,
	output reg SRAM_UB_N,
	output reg SRAM_LB_N,
);

always @(posedge clk)
begin
	assign SRAM_CE_N = ! chipselect;
	assign SRAM_ADDR = address;
	assign SRAM_WE_N = ! write;
	assign SRAM_OE_N = ! read;
	assign SRAM_LB_N = ! byte_enable[0];
	assign SRAM_UB_N = ! byte_enable[1];

	assign read_data = (read) ?SRAM_DQ:16'bz;
	assign SRAM_DQ = (write) ?write_data:16'bz;
end
endmodule
```

Question 7:

```
#include "system.h"
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int i;
//-----------------------------------char test-------------------------------//
```

```
char read[5];
char write[5] = {'1', '2', '3', '4', '5'};
char *mem = (char *)SRAM_CONTROLLER_0_BASE;
void test_char(){
	for (i=0; i<5; i++){
		*mem = write[i];
		printf("writing: %c \n", *mem);
		mem++;
	}
	return;
}
//-----------------------------------short test-------------------------------//
short read[5];
short write[5] = {'1', '2', '3', '4', '5'};
short *mem = (char *)SRAM_CONTROLLER_0_BASE;
void test_short(){
	for (i=0; i<5; i++){
		*mem = write[i];
		printf("writing: %c \n", *mem);
		mem++;
	}
	return;
}
//-----------------------------------int test---------------------------------//
int read[5];
int write[5] = {'1', '2', '3', '4', '5'};
int *mem = (char *)SRAM_CONTROLLER_0_BASE;
void test_int(){
	for (i=0; i<5; i++){
		*mem = write[i];
		printf("writing: %c \n", *mem);
		mem++;
	}
	return;
}
```