INDIAN INSTITUTE OF TECHNOLOGY, DHARWAD

Course Project Assignments

1. The specifications given here are to be taken as guidelines.
2. Projects are inter-related. Read the description for all before starting your design.
3. Registers B,C,D,E,H,L,M and A will be identified by 3 bit codes 000, 001, 010, 011, 100, 101, 110 and 111 respectively. Note that M is not a register, actually - it is the data fetched by the Bus Interface unit from the address contained in HL.
4. Register pairs BC, DE, HL and PSW will be encoded as 00, 01, 10 and 11 respectively. (11 will sometimes represent SP, depending on context).
5. Use signals of type std_logic or std_ulogic as appropriate. Do not assume any external math packages etc. Descriptions must be synthesizable.
6. A test bench should illustrate the working of the circuit.
7. Design will need to be ported to an FPGA board.

Proj–1 Design the bus interface unit for the processor. This unit will be connected to the regular 8085 bus on the outside and a 16 bit address bus, 8 bit data bus and the following control signals on the inside:

a) Status bits S0 and S1 and $IO/\overline{M}$, which encode the kind of IO to carry out. (Instruction fetch, data read, data write etc.) These bits will also appear on the outside.

b) Start and Done signals. Start is an input to your circuit and is a positive pulse which is one crystal clock wide. Done should be an output from you indicating the completion of operation. This should also be one crystal clock wide.

c) An output which is true if an interrupt is seen on the bus during the current cycle. This is also a pulse which is one crystal clock wide.

Proj–2 Design the instruction processor. This is an FSM which fetches the next instruction from an address given by the contents of the 16 bit register 'PC', and stores it in the instruction register. This circuit will also execute all instructions which update 'PC'. (Unconditional and conditional jump, call and ret instructions, for example).

For other instructions, issue a 'DoData' pulse to the data processor described below. You should use the project on Bus Interface Unit for memory read/write operations. Use a simple behavioural model of the same for your test bench.

Proj–3 Design the data processor. The data processor should implement the ALU and should carry out all data movement and ALU operations. This includes PUSH and POP operations involving the stack processor. You should use the project on Bus Interface Unit for memory read/write operations. Use a simple behavioural model of the same for your test bench.

Proj–4 Design the interrupt processor. This should implement all hardware and software interrupts. (Refer to any 8085 book and the data sheet). You should use the project on Bus Interface Unit for memory read/write operations. Use a simple behavioural model of the same for your test bench.