Project 5 - 8086 Board Design

Collin Mood, Ryan Bugianesi

CMPE 310

Prof. Smith

Hardware:

To appropriately buffer and de-multiplex the joint Address/Data bus on the 8086, a series of buffer chips and latches are used. Two 373's are used across the bottom 16 bits to pull the address bus off of the data bus. These also function as buffers for the bottom 16 bits of the address bus. The top four bits are pulled directly from the 8086, and buffered with an additional 373. The address bus buffers are enabled by the ALE pin from the 8086,while the data bus buffers are determined by the DT/~R, and ~DEN pins, both sent to two LS245's. The bottom four control pins are buffered through an LS244 chip. For decoding the addressing for most chips, we used PAL16L8s and a combination of address pins and processor signals to achieve the correct chip selects and the correct addressing for each chip. The SRAM was slightly more complex since it required two additional chips for a total of four, which caused us to have to consider not only a high and low bank, but also even and odd addressing in each bank. This was solved using the same 16L8 and logic using A0 to decode even and odd addresses between the two chips in the high and low bank.

Design Choices:

We chose to use a PAL16L8 chip to decode and address both the 256K of CMOS flash and the 128K of SRAM. The CMOS flash is decoded from the top address of the address range (FFFFFH) to location E0000H. This uses 17 of the 20 address lines, leaving the last three open. Since the memory is supposed to be located in the highest address of the microprocessor, we are using the three leftover pins for logic in the 16L8. A0 from the address bus is renamed to the ~BLE pin in the CMOS-28F010.sch file. This is because A0 will select either the high bank or the low bank in combination with the ~BHE signal from the 8086 processor itself. The M/~IO pin is used in the logic to enable and disable the flash memory when the microprocessor is reading/writing to memory or IO. The WE and ~OE pins are used to indicate a read operation or a write operation respectively. The SRAM implementation differs slightly from the design described above because it was located from the lowest address available in the microprocessors address range (00000h) and holds 64K addresses. Each CY7C199 stores 32K address locations, each one byte in size, so two chips were needed to get the 64K address locations required in the project documentation. The project also specified a high and low bank, so the processor could write and read 8 or 16 bit data. We used another PAL16L8 and a combination of the address pins to decode the chip select of the four CY7C199 SRAM chips.

Hierarchical pins were placed inside of each sheet that required part or all of a bus. In the 8086, individual pins were placed inside of the sheet, and the data, address, and control busses were created on the main sheet outside to allow for branching off of any necessary bus partially, rather than sending the entire bus to each component that only required a few pins from it.

Workload Division:

The workload was split relatively organically, Collin working on the multiplexing and buffering of the buses from the 8086, while Ryan was tasked with decoding the address pins for the SRAM. Ryan also decoded the address ranges for the CMOS flash memory. Necessary pins were determined by Ryan and placed into appropriate 16L8 PLDs, with some input from Collin concerning the logic required for the address ranges, due to the relative ease of demultiplexing compared to address decoding.