Memory interfacing in 8085 pdf

Memory is an integral part of the microprocessor system, and in this section we'll discuss how to interact with a microprocessor. Memory interaction in 8085 is used to access memory quite often to read the codes of instructions and data stored in memory. These reading/recording operations are controlled by control signals. The microprocessor activates these signals when it wants to read and write in memory. In the last section, we've already seen reading and memory cycles, as well as RD, WR, and IO/M state signals for reading/writing. In the next section we will see the structure of memory and its requirements, the concepts in the memory of Interfacing in 8085 and interconnected examples. Memory Structure and Requirements: As mentioned earlier, read/write memories consist of an array of registers in which each register has a unique address. Memory size N x M, as shown in the pic. 4.13 (a) where N is the number of registers and M is the length of the word, in the number of bits. Basic concepts in Interfacing's memory: For the memory of Interfacing in 8085, the following important points should be kept in mind. The 8085 microprocessor can access 64Kbytes memory because the address bus is 16-bit. But it is not always necessary to use the full address area of 64Kbytes. The total memory size depends on the app. Typically, EPROM (or EPROMs) is used as program memory and RAM (or RAMs) as data memory. Using EPROM and RAM, the total address area of 64Kbytes is shared by them. The memory capacity of the program and the memory of the data depends on the application. You don't always need to choose 1 EPROM and 1 RAM. We may have several EPROMs and several RAMs as required. We can place EPROM/RAM anywhere in the full 64 Kbytes address

space. But the program's memory (EPROM) should be located at 0000H, Since the reset address of 8085 microprocessor is 0000H. It is not always necessary to find EPROM and RAM in sequential memory for example: If the EPROM display is from 0000H to OFFFH, it is not necessary to find RAM from 1000H. We can find it anywhere between 1000H and FFFFH. Where to find a memory component depends entirely on the application. Memory Interaction Requires: Select Chip Identify Register To Include the Appropriate Buffer. The microprocessor system includes memory devices and I/O devices. To communicate with memory devices or v-o, you need to decipher the address from the microprocessor. This allows each device (memory or I/O) to be available on its own. The next section describes common methods of decoding Address decoding methods : Absolute decoding/complete decoding of linear linear Decoding absolute decoding: In absolute decoding technique, ever higher address lines are deciphered to select a memory chip, and the memory chip is selected only for specified levels of logic on these high-order address lines; no other levels of logic can choose a chip. Fig. 4.14 shows the interaction of memory in 8085 with absolute deciphering. This method of handling is usually used in large memory systems. Linear decoding: In small systems, decoding logic equipment can be eliminated using separate high-order address lines to select memory chips. This is called linear decryption. Fig. 4.15 shows the address of RAM using linear decoding technique. This method is also called partial decryption. This reduces the cost of deciphering the scheme, but has a shortage of several addresses (shadow addresses). Fig. 4.15 shows the address of RAM using linear decoding technique. The A15 address line is directly connected to the EPROM chip selection signal and after the inversion is connected to the RAM chip selection signal. Therefore, when the status of the A15 line is zero, EPROM is selected and the status of the A15 line one of RAM is chosen. The status of other address lines is not considered because these address lines are not used to generate chip selection signals. Wait State Generator: We've seen that waiting conditions can be entered in machine cycles using the READY signal for the slower memory device interface. In this section, we'll see diagrams for the waiting state generator and the application of these diagrams for the slower memories interface. Fig. 4.22 shows the diagram for the waiting state generator. This diagram generates one waiting state in each machine cycle. The chain operation is below. The T1 activates the ALE signal, making the output of the first flip-flop high, suggesting that RESET (R) is one of them. In the next pulse of the watch, i.e. at the next low and high transition of the CLKOUT 8085 signal, the output of the second flip-flop D goes low. Low exit no1 initiates 8085 to enter the standby state and resets the first D flip-flop, making it no0 low, as No.1 is connected to the finished 8085 input and to reset the entry of the first flip-flop D. At the next low to high-jump CLKOUT, the second D flip flop goes high, making READY contact high and inactivating the input of the first flip-flop. Fig. 4.23 shows the timing of wave forms for this chain. This scheme can be modified to enter the waiting states for a particular machine cycle. The modified waiting state generator is shown in the pic. 4.24. The modified scheme will generate one WAIT state for any machine cycle for which wait cycle machine output to choose the logic 0 scheme. We are quite a few times in this course 8085 is about how 8085 reads and writes data from or to memory. But we know that microcontrollers that have a certain amount of built-in memory, microprocessors do not have their own memory, except for a few registers. So where does 8085 read and write the data? Since the 8085 doesn't have significant internal memory, we need to attach external memory chips. How do I connect the memory chip to the 8085 and manage the interaction between the two chips? This is what we are going to learn in this post. We will address the problem and address it step by step, studying the concepts in the process as we move forward. Memory support in the 8085An 8085 microprocessor has a 16-bit address bus. Each bit can take a value of 0 or 1. Thus, the total number of addresses that can be created on a 16-bit address bus will be 256. And each unique address refers to a memory block containing 8 bits or 1 space fore fore. Thus, we can say that the 8085 can support a memory chip of up to 64 kB. We can interface the memory chip size smaller than that too. In addition, we can interface multiple memory chips for a single 8085 microprocessor, as long as their cumulative size does not exceed 64 kB. Let's learn how to achieve all this. Memory chips of different types and sizesMemory chips come in different types and with different storage capabilities. Broad classification of memory chips based on their reading and recording capabilities: RAM (Random Access Memory): We can read as well as write data on this type of memory. This type of chip has contacts for memory and memory reading to write signals. ROM :Read only memory: As the name suggests, we can only write data on this type of memory chip. This type of chip has a pin only for the memory reading signal. Now, you should be surprised that the microprocessor read with ROM if the data can't be written on it. But the data can be written on it using some special methods. This type of memory is used to store programs, while RAM is used to store data. ROM also has two types: EPROM (Erasable Programmable Read-Only Memory): EPROM content is erased by ultraviolet rays. The data is written on it optically. EEPROM (Electronic Erasable Programmable Memory Only Reading): As the name implies, the data is written and deleted on this type of ROM electronically. You can learn more about all types of digital memories here. Memory chips come in different sizes. The image below shows what these numbers mean in the memory chip capacity specification (size). Now let's learn from the example how external RAM and external ROM chips can interact with 8085. Let's take on the problem of memory interaction and solve it as we learn about it. Memory interaction - StatementInterface 1kB EPROM and 2kB of RAM with microprocessor 8085. The address, set aside 1 KB EPROM, should be from 2000H to 22FFH. You can a range of addresses of your choice to 2 kB RAM. The first step to solving this problem is to understand the contacts of this particular memory chip. The RAM and ROM pin chart has the same contacts, with the exception of the WR pin, which is present in RAM and is not present in the ROM. Let's see the pins one by one. Data Contacts: Since each memory location stores eight bits, there are eight lines of D0-D7 data connected to the memory chip. Address contacts: The number of address contacts depends on the size of the memory. In this case, the memory size of 1 kB x 8 will have 210 different memory places. Thus, it will have ten A0 address lines up to the A9. Similarly, 2 kB of RAM will have 211 different memory sites. So, there are 11 address lines A0-A10.CS: When this pin is on, the memory chip knows that the microprocessor is talking to it and reacting accordingly. We have to generate this signal for each of the chips according to the range of addresses assigned to them. Chip Select (CS) pin is used to do this. OE Contact: When this active-low output turn on the pin enabled, the memory chip can exit the data into the data bead. WR Contact: After activating this active-low memory write a contact, the data on the data booth is written on the memory chip in the place specified by the address beads. VCC and GND pins: These contacts serve the purpose of powering ICs. For simplicity, we will not show these pins on the diagram. There are three types of buses in 8085 - Bus Address, Bus Data, and Bus Control. Each of these buses will be connected to the memory chip. Connecting control signalsIn memory chips there are two pins for control signals - OE (Output Enable) and WR (Memory Write). They will be connected to control signals generated by a decoder of 3 to 8. To read about signal control generation, you can read our post on Demultiplexing Bus and Signal Control. The control signal generation scheme is shown below. Four control signals are generated by the input of WR, RD and IO/M signals from 8085 to 3:8 of the decoder - IOR, IOW, MEMR and MEMW. Since we're dealing with memory, we just need MEMR and MEMW signals. When reading from a memory chip, its output should be turned on. Thus, MEMR will be connected to the OE pin. Similarly, to record on the memory chip, the MEMW will be connected to the WR RAM pin. After completing these two connections, we did with the control signals, except for CS. We'll deal with this in a bit. Data Bus interfacingThere consists of eight lines, including a bead of data of both 8085 and memory chips. Data boos interaction is the simplest part. We simply connect the appropriate lines (D0-D7 from 8085) to pins (D0-D7 memory chip). The Interfacing bus address has 2kB RAM with 11 address lines. So, the first 11 lines Bus 8085 will be connected to the appropriate 2kB RAM address line. Similarly, the first 10 lines of the 8085 address bus will be connected to the respective 1kB EPROM lines. The remaining address lines will be used to generate the Chip Choice (CS) signal. It's a bit tricky, but it's the most important part of solving the problem. Let's act step by step and create an intuition on how to generate a chip to select a signal for the memory of this size and this range of addresses. Let's affect the starting and last address of 1kB EPROM. The A15 is the most significant, and the A0 is the least significant bit. The range of addresses for EPROM placement is from 2000H to 22FFH (as of this issue.) Translation of these to binary: 2000H and 0011 00000 00022FFH - 0011 0011 1111 1111 Address the bit numberA15A14A13A12A11A11111110 A9A8A7A6A5A4A3A2A1A0Starting address011000000000Ending Address0011111111111111111111111111 that ten bits from A0 to A9 change. These ten bits are directly related to the address line of the memory chip. These ten bits take a value of 0 or 1 to form addresses. The first address is 00,000,000, and the second address is 00,000,000,0001, the third address is 000,000 0010, and so on. The last address will be 11 1111 1111. Meanwhile, the A11 bits to the A15 do not change or affect the address process inside the memory chip. Thus, we can conclude that the values of the A15-A11 bits (0011 00) data in the table above are in a unique, unchanged configuration for this memory chip. If at least one of these bits changes, the address won't belong to that memory chip. So we can use these A15-A111 values to unequivocally identify this memory chip, which is exactly what cs signal should do. You could say that when A15 and A14, A11, A10 and A13 and A12 1, then our memory chip should be selected. Now we need to develop a logic for generating the CS signal. As a result, boolean cs equation will be: CS and supplement (A15 . A14 . A13 . A12 . A11 . A10) This equation can be implemented with NAND Gate. The final chip to choose the logic for 1kB EPROM is illustrated below. Now we have to generate a chip select signal for the second memory chip, which is 2kB RAM. The process is very similar and different from the previous two ways: memory size is different. So, there are 11 address lines instead of 10.We don't give the address range here. We are given the freedom to make our own decisions. As in the previous case, we connect the first 11 8085 microprocessor address lines to the 11th 2kB RAM address line. These bits will take values 0 and 1 and will generate 2 and 1024 different addresses. The address bits of the A10-A0 will range from 000,000,000 to 111 1111 1111.And what about the rest of the bit address? Well, they don't any role in solving memory in this 2kB RAM. In this way, we can correct them to a certain value without affecting anything. Let's fix them to 0000 0. Thus, the range of addresses for this chip becomes 0000 0000 0000 000 0000 to 0000 0111 1111 1111. In hexadecimal, the address range will be from 0000H to 07FFH. Address bit numberA15A14A13A12A111A111A10A8A7A6A4A4A2A1A0Start address00000000000Ending Address000000000011111111111111111111111111111111111111111111 We use a similar technique here. We use the remaining A15-A11 bits to unambiguously identify this chip, i.e. to generate the chip selection signal. Thus, the boolean equation will complement (A15). A14 . A13 . A12 . A11) The implementation of this equation using NAND Gate to generate the CS signal is shown in the following image. The final scheme Now we have the chip to choose the logic and decided all the connections, it's time to complete the scheme. The entire external memory interphasization circuit can be broken down into five different parts:8085 Demultiplexing Microprocessor address/busGeneration data signal-processing chip select signalsMemory chipsIn the images below show the final diagram with all five parts listed above integrated into a single diagram. Only connections appear on the first chart. The next diagram shows the different subsections of the chain. The chart above summarizes the entire process of external memory interpretation with the 8085 microprocessor. If you have any questions on this topic, drop the comment below and we'll get back to you. You. memory interfacing in 8085 microprocessor ppt. memory interfacing in 8085 examples. memory interfacing in 8085 examples pdf. memory interfacing in 8085 microprocessor pdf. memory interfacing in 8085 problems. memory interfacing in 8085 microprocessor examples. memory interfacing in 8085 microprocessor notes. memory interfacing in 8085 geeksforgeeks