FSM Control for I/O

Example Show how the control for an output transaction to an I/O device from memory (as in Figure 8.7) can be implemented as a pair of finite state machines.

Answer Figure 8.11 shows the two finite state machine controllers that implement the handshaking protocol of Figure 8.10.

If a synchronous bus can be used, it is usually faster than an asynchronous bus because of the overhead required to perform the handshaking. An example demonstrates this.

Performance Analysis of Synchronous versus Asynchronous Buses

Example We want to compare the maximum bandwidth for a synchronous and an asynchronous bus. The synchronous bus has a clock cycle time of 50 ns and each bus transmission takes 1 clock cycle. The asynchronous bus requires 40 ns per handshake. The data portion of both buses is 32 bits wide. Find the bandwidth for each bus when performing one-word reads from a 200-ns memory.

Answer First, the synchronous bus, which has 50-ns bus cycles. The steps and times required for the synchronous bus are as follows:

1. Send the address to memory: 50 ns
2. Read the memory: 200 ns
3. Send the data to the device: 50 ns

Thus, the total time is 300 ns. This yields a maximum bus bandwidth of 4 bytes every 300 ns, or

$$\frac{4 \text{ bytes}}{300 \text{ ns}} = \frac{4 \text{ MB}}{0.3 \text{ seconds}} = 13.3 \frac{\text{MB}}{\text{second}}$$

At first glance, it might appear that the asynchronous bus will be *much* slower, since it will take seven steps, each at least 40 ns, and the step corresponding to the memory access will take 200 ns. If we look carefully at Figure 8.10, we realize that several of the steps can be overlapped with the memory access time. In particular, the memory receives the address at the end of step 1 and does not need to put the data on the bus until the beginning of step 5; steps 2, 3, and 4 can overlap with the memory access time. This leads to the following timing:

Step 1: 40 ns

Steps 2, 3, 4: maximum (3 x 40 ns, 200 ns) = 200 ns

Steps 5, 6, 7: 3 x 40 ns = 120 ns

Thus, the total time to perform the transfer is 360 ns, and the maximum bandwidth is

$$\frac{4 \text{ bytes}}{360 \text{ ns}} = \frac{4 \text{ MB}}{0.36 \text{ seconds}} = 11.1 \frac{\text{MB}}{\text{second}}$$

Accordingly, the synchronous bus is only about 20% faster. Of course, to sustain these rates, the device and memory system on the asynchronous bus will need to be fairly fast to accomplish each handshaking step in 40 ns.

Even though a synchronous bus may be faster, the choice between a synchronous and an asynchronous bus has implications not only for data bandwidth but also for an I/O system's capacity in terms of physical distance and the number of devices that can be connected to the bus. Asynchronous buses scale better with technology changes and can support a wider variety of device response speeds. It is for these reasons that I/O buses are often asynchronous, despite the increased overhead.

Increasing the Bus Bandwidth

Although much of the bandwidth of a bus is decided by the choice of a synchronous or asynchronous protocol and the timing characteristics of the bus, several other factors affect the bandwidth that can be attained by a single transfer. The most important of these are the following:

1. *Data bus width*: By increasing the width of the data bus, transfers of multiple words require fewer bus cycles.
2. *Separate versus multiplexed address and data lines*: Our example in Figure 8.8 used the same wires for address and data; including separate lines for addresses will make the performance of writes faster because the address and data can be transmitted in one bus cycle.

Performance Analysis of Two Bus Schemes

Example Suppose we have a system with the following characteristics:

1. A memory and bus system supporting block access of 4 to 16 32-bit words.
2. A 64-bit synchronous bus clocked at 200 MHz, with each 64-bit transfer taking 1 clock cycle, and 1 clock cycle required to send an address to memory.
3. Two clock cycles needed between each bus operation. (Assume the bus is idle before an access.)
4. A memory access time for the first four words of 200 ns; each additional set of four words can be read in 20 ns. Assume that a bus transfer of the most recently read data and a read of the next four words can be overlapped.

Find the sustained bandwidth and the latency for a read of 256 words for transfers that use 4-word blocks and for transfers that use 16-word blocks. Also compute the effective number of bus transactions per second for each case. Recall that a single bus transaction consists of an address transmission followed by data.

Answer For the 4-word block transfers, each block takes

1. 1 clock cycle that is required to send the address to memory
2. $\frac{200 \text{ ns}}{5 \text{ ns/cycle}} = 40$ clock cycles to read memory
3. 2 clock cycles to send the data from the memory
4. 2 idle clock cycles between this transfer and the next

This is a total of 45 cycles, and 256/4 = 64 transactions are needed, so the entire transfer takes $45 \times 64 = 2880$ clock cycles. Thus the latency is 2880 cycles $\times$ 5 ns/cycle = 14,400 ns. The number of bus transactions per second is

$$64 \text{ transactions} \times \frac{1 \text{ second}}{14{,}400 \text{ ns}} = 4.44\text{M transactions/second}$$

The bus bandwidth is

$$(256 \times 4) \text{ bytes} \times \frac{1 \text{ second}}{14{,}400 \text{ ns}} = 71.11 \text{ MB/sec}$$

For the 16-word block transfers, the first block requires

1. 1 clock cycle to send an address to memory
2. 200 ns or 40 cycles to read the first four words in memory
3. 2 cycles to send the data of the block, during which time the read of the four words in the next block is started
4. 2 idle cycles between transfers and during which the read of the next block is completed

Each of the three remaining 4-word blocks requires repeating only the last two steps.

Thus, the total number of cycles for each 16-word block is $1 + 40 + (2 + 2) = 57$ cycles, and $256/16 = 16$ transactions are needed, so the entire transfer takes, $57 \times 16 = 912$ cycles. Thus the latency is 912 cycles $\times$ 5 ns/cycle = 4560 ns, which is roughly one-third of the latency for the case with 4-word blocks. The number of bus transactions per second with 16-word blocks is

$$16 \text{ transactions} \times \frac{1 \text{ second}}{4560 \text{ ns}} = 3.51\text{M transactions/second}$$

which is lower than the case with 4-word blocks because each transaction takes longer (57 versus 45 cycles).

The bus bandwidth with 16-word blocks is

$$(256 \times 4) \text{ bytes} \times \frac{1 \text{ second}}{4560 \text{ ns}} = 224.56 \text{ MB/second}$$

which is 3.16 times higher than for the 4-word blocks. The advantage of using larger block transfers is clear.

Elaboration: Another method for increasing the effective bus bandwidth when multiple parties want to communicate on the bus is to release the bus when it is not actually used for transmitting information. Consider the example of a memory read that we examined in Figure 8.10. What happens to the bus while the memory access is occurring? In this simple protocol, the device and memory continue to hold the bus during the memory access time when no actual transfer is taking place. An alternative protocol, which releases the bus, would operate like this:

1. The device signals the memory and transmits the request and address.
2. After the memory acknowledges the request, both the memory and device release all control lines.
3. The memory access occurs, and the bus is free for other uses during this cycle.