DMA on the SBC-188 v3

Introduction:

The RetroBrew Computer ECB bus has evolved from the Kontron Z80 bus. The latter used only the A and C rows (64-lines) on the 96-pin DIN 41612 connector. As the ECB bus evolved, pins on the B row lines began to be used, until virtually all of the 96 lines on the connector have dedicated uses.

The address bus has been expanded from 16 lines to 32 lines and the data bus has been expanded from 8 lines to 16 lines. 8 interrupt lines are defined now, and the bus is designed for processors from the Z80 up to the MC68030.

Z80 ECB background:

Signals that tristate on the Z80 ECB bus:

- Address (64K, 16 bits)
- Data (1 byte, 8 bits) (only a proposed NS32000 cpu uses 16 bits)
- MREQ# – memory request
- IORQ# – input/output request
- RD# – read
- WR# – write

Signals that control bus takeover from the CPU:

- BUSRQ# – bus request
- BUSAK# – bus request acknowledge

Although external DMA was thought of on the original Z80 SBC and follow on revisions, (1) I am pretty sure no one attempted it, and (2) I am pretty sure it did not work. With this background, DMA was not a consideration on the Z180 Mark IV board, nor on the Motorola 68000 boards: MC68008 and KISS-68030.

DMA considerations:

On the Z80, the tristate signals are those mentioned above. On the 80C188, one more signal should be added to this list: DT/R (Data Transmit/Receive) which indicates the direction of data transfer from the CPU; hence, the direction of data transfer across the ECB bus, if the transfer is to/from external memory or I/O. DT/R is used to set the data flow direction through the LS245 data transceiver. It should be noted that this chip is only enabled if the memory or I/O reference is across the ECB bus, never for local CPU board operations.

SBC-188 version 3:

Version 3 of the SBC-188 very carefully makes sure that when the ECB bus has been acquired by an external board, signals MREQ#, IORQ#, RD#, WR#, & DT/R have tri-stated. The signals are now supplied by the ECB bus master to control access to memory. Due to an 80C188 limitation, I/O peripherals on the board are not accessible; only main memory is accessible (1Mb). Similarly, the design of the CPU board will not allow access to ROM or external memory (4MEM or VGA3); only the R/W 640K + high memory blocks are accessible. The limitation is due to the fact that ROM address decode, External Memory address decode, and I/O select decode are all done inside the high-integration CPU chip. This is not seen as any limitation; the only access of interest should be the CPU board R/W memory.

The DMA protocol:

The access protocol operates as follows:

1. BUSRQ# is asserted, and the xDMA controller has control of the ECB bus when it receives BUSAK#. (Below, “It” refers to the external DMA controller, xDMA.)
2. It asserts 20 address lines, now tri-stated by both the CPU & the address latches. During the transfer of control, all control signals are held high by pull-up resistors.
3. It sets DT/R to indicate the direction of data transfer.
4. It may then, or at the same time, assert MREQ# to indicate which memory location to select.
5. If a write to memory is the operation, it may now assert the data to be written.
6. It then asserts either RD# or WR#.
7. It latches data out of memory before or at the end of RD#; and must maintain data to be written until after the end of WR#, at which time the SRAM memory latches the data.
8. The release of MREQ# will terminate the SRAM chip select at the address on the bus.
9. It releases DT/R before terminating the external DMA access, most likely at the end of MREQ#.
10. The xDMA controller has the option of releasing the ECB bus at this point, or performing another data transfer by updating the address lines. If MREQ# is not de-asserted between transfers, then enough time for the memory to select the new address must elapse before the next RD# or WR#.
11. All signals should be tri-stated before BUSRQ# is de-asserted to return control to the CPU. The CPU signals the return of the bus to its control by de-asserting BUSAK#. During the transfer of control, bus control signals are held high by pull-up resistors.

If the CPU board in question supports external DMA access to peripheral chips on the CPU board, the above protocol is followed exactly by substituting IORQ# where MREQ# mentioned. NOTE: as mentioned above, the SBC-188 does not allow external DMA access to CPU board peripherals; only the CPU-internal DMA controllers have such access.

Other considerations:

The expansion of the Z80 protocol to specify use of DT/R (a row B signal on the 96-DIN connector) is in line with 80C188, MC68000, and NS32000 usage. It is generally asserted very early in a bus cycle, often before address lines are latched. It avoids the problem of multiple data drivers driving each others' outputs simultaneously.

DT/R becomes a tricky signal. It represents the direction of data flow across the bus; it does NOT represent whether a read or a write cycle is to take place. When the CPU has control of the bus, DT/R is high on memory writes (which could be a write to an external device out on the ECB bus), and low on memory reads. For the integrated DMA controllers on the 80C188 chip, DT/R has the same sense as CPU access cycles, since it is generated within the highly integrated processor chip.

DT/R retains the same sense for an external DMA controller. But the sense with respect to read or write operations is opposite. The sense for data transfer across the ECB bus, does, however, remain the same. On an xDMA read operation, data transfer is out from memory, out from the CPU board, and across the ECB bus. Hence DT/R must be high to indicate the data transfer direction. Similarly, DT/R must be low on an xDMA write operation, since data is flowing across the ECB bus into (onto?) the CPU board where the memory is located.

The xDMA controller should think of DT/R as the same signal as RD/WR#. If such signal name is used to drive the ECB bus DT/R pin, then the xDMA controller will be getting the bus direction set correctly for its operations.

Coda:

As there are no RetroBrew boards that attempt DMA across the bus at the moment, I just hope to start everyone thinking seriously about the topic.

--John Coffman
08-Feb-2020