MPC860 Interface with Generic MC68000 Bus

The main point of this document is to demonstrate an interface between an MPC860 and a generic MC68000 peripheral. It should be noted that this interface only supports read or write accesses initiated by the MPC860 to the peripheral. For the purpose of this document the following parameters are assumed:

- Interface is between an MPC860 and a lower speed MC68000 peripheral.
- MC68000 peripheral is a 16 bit port size that is defined as part of the upper data bus.
- The MC68000 peripheral is not a bus master.
- The MPC860 user programmable machine (UPM) is programmed to generate the proper timing of the signals.
- General purpose input/output pins available from the MPC860 are used to accommodate function code programming of the MC68000. Alternatively, external logic could be used to make the existing signals behave like MC68000 function code signals.

Programming the Option, Mode and Base Registers

The option and base registers of the MPC860 must be programmed as follows for this type of interface. The option and base registers must be programmed for a 16 bit (no-parity) non-bursting port. The following tables have been appended from section 15 of the MPC860 Manual.

Table 15-14. Base Register

Bits	Mnemonic	Description	Function
20-21	PS(0:1)	**Port Size.** This field specifies the port size of the memory region.	10 = 16 bit port size
22	PARE	**Parity Enable.** This bit is used to enable of the parity charity checking on this bank.	0 = disable parity
24-25	MS(0:1)	**Machine Select.** This field specifies the machine selected for the memory operations handling.	10 = U.P.M.A or 11 = U.P.M.B
31	V	**Valid Bit.** This bit indicates that the contents of the base register and option register pair are valid. The CS signal does not assert until the V bit is set. Note: An access to a region that does not have the V bit set may cause a bus monitor time out. Following a system reset, the V bit is set in BR0.	

Table 15-15. Option Register

Bits	Mnemonic	Description	Function
20	SAM	**Start Address Multiplex.** This attribute determines how the address is output on the first cycle of an external memory access read or write when the memory access is handled by UPMA or UPMB.	0 = Address pins are the address requested by the internal master. Disable address multiplexing.
23	BI	**Burst Inhibit.** This attribute decides whether or not this memory bank supports burst accesses. In a non-burst case, the memory controller drives the BI signal active when accessing this memory region. Following a system reset, the BI bit is set in OR0.	1= Drive BI asserted. The bank does not support burst accesses.

Table 15-16. Machine A Mode Register

Bits	Mnemonic	Description	Function
8	PTAE	**Periodic Timer A Enable.** This bit allows the periodic timer A to request service. Note: Following a system reset, the PTAE bit is reset.	0 = Periodic timer A is disabled
19	GPL_A4DIS	**GPL_A4 Output Line Disable. This** bit determines if the UPWAITA*/ GPL_A4* pin will behave as an output line controlled by the corresponding bits in the UPMA array (GPL4A*). Note: Following a system reset, the GPL_A4DIS bit is set.	1 = UPWAITA/GPL_A4 behaves as a UPWAITA when: The G4T4/DLT3 bit in the UPMA is interpreted as DLT3. The G4T3/WAEN bit in the UPMA is interpreted as WAEN.

Programming the Signals for a Successful MPC860 and MC6800 Generic Bus Interface:

For this interface the MPC860 signals are programmed to communicate with the MC68000 Bus. With examination of the timing specifications, the MPC860 can be programmed to interface with any of the MC68000 peripherals. In each case the UPM can be programmed to accommodate different part speeds.

The active low MPC860 chip select signal is programmed to the MC68000 address strobe. The chip select's assertion can be interpreted as a MC68000 address strobe to emulate the MC68000 bus operation. Because you can directly manipulate chip select timing via the UPM, you can reproduce the timing of the MC68000 AS* signal.

The active low signals BS[0] and BS[1] can be programmed for the upper and lower data strobe signals of the MC68000. BS[0] and BS[1] correspond with the byte lanes of the upper half of the data bus. The timing of these BS* signals can also be directly manipulated via the UPM, enabling emulation of the MC68000 UDS* and LDS* signals.

The Read/Write signal of the MPC860 can directly communicate with the same signal on the MC68000. The timing is not a problem since the read/write signal is asserted directly after the transfer start signal.

When programming the function codes of the MC68000 bus, strict attention must be paid to the function code outputs. For the following explanation please refer to page 13-35 of the *MPC860 User's Manual* and page 3-9 of the *MC68000 User's Manual*. For the purpose of this example, UPM general purpose IN/OUTPUT lines are used to perform as function code signals. Timing of these signals is not critical, they merely must stay constant throughout the access to the MC68000 peripheral. However, only one function code may be programmed. It is also possible to generate the function codes in the hardware. Please see Figure 1-1 through Figure 1-3 for a detailed diagram of the interface and the function code hardware. Note that this configuration does not support the MC68000 IACK signals. This only supports the following four function codes: supervisor program, supervisor data, user program and user data.

The DTACK* signal must be handled by an external OR gate. This will generate an UPWAIT signal until the AS* and DTACK* signals are asserted. The UPM output signals will freeze as long as UPWAIT is asserted. The UPWAIT signal becomes de-asserted when the DTACK* signal is asserted by the MC68000 peripheral. Figures 1-1 and 1-3 show the complete interface and timing diagram.

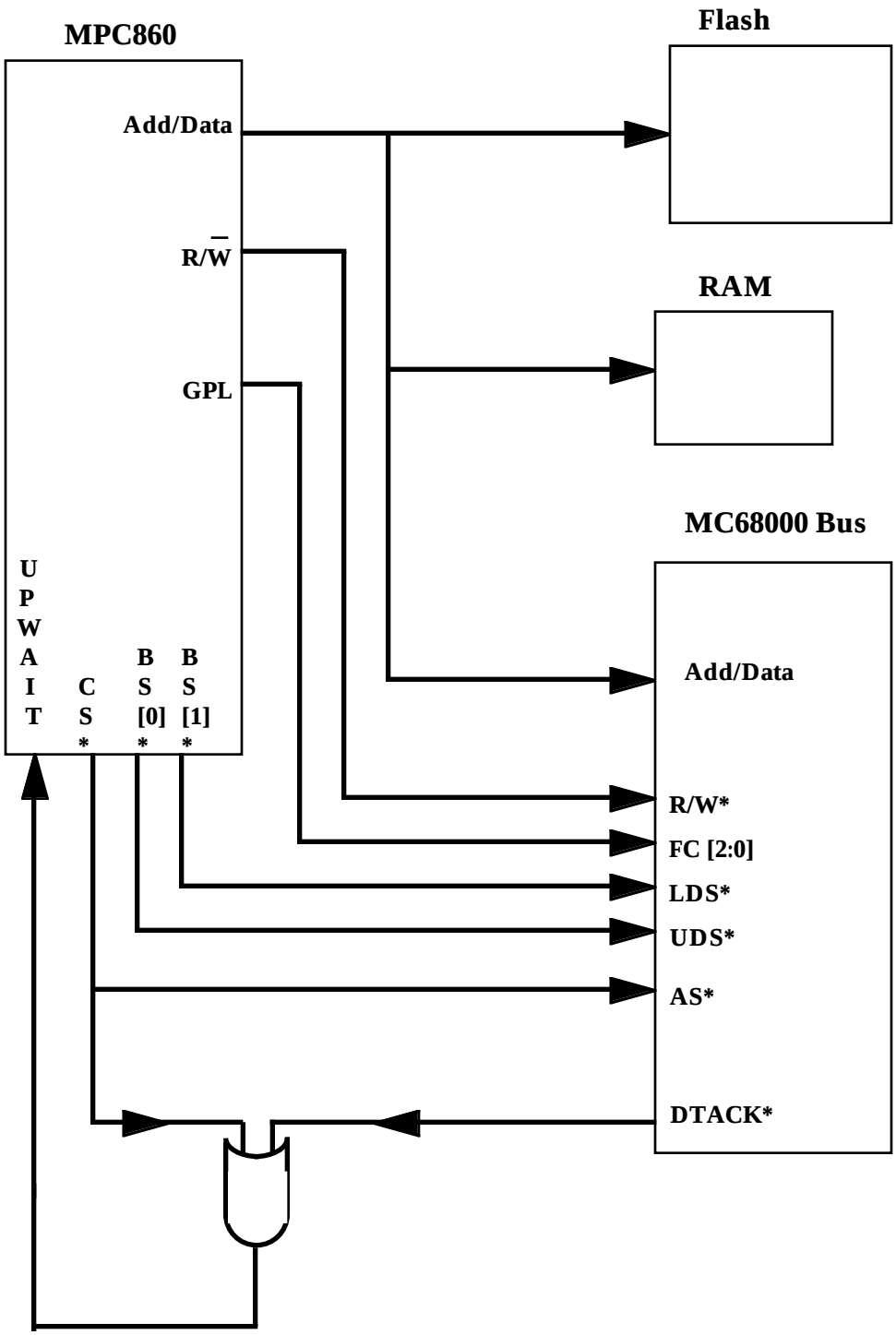
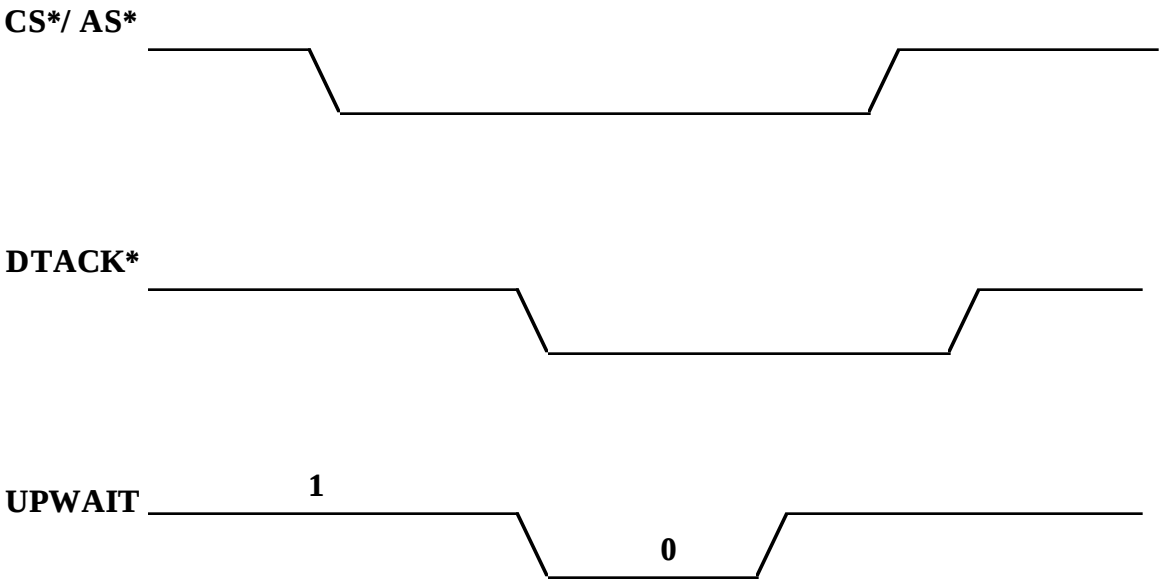


Figure 1. MPC860 Interface with Generic 68K Bus



UPWAIT = (CS* • DTACK*)* = CS + DTACK

Figure 2. MPC860 Interface with Generic 68K Bus Timing Diagram

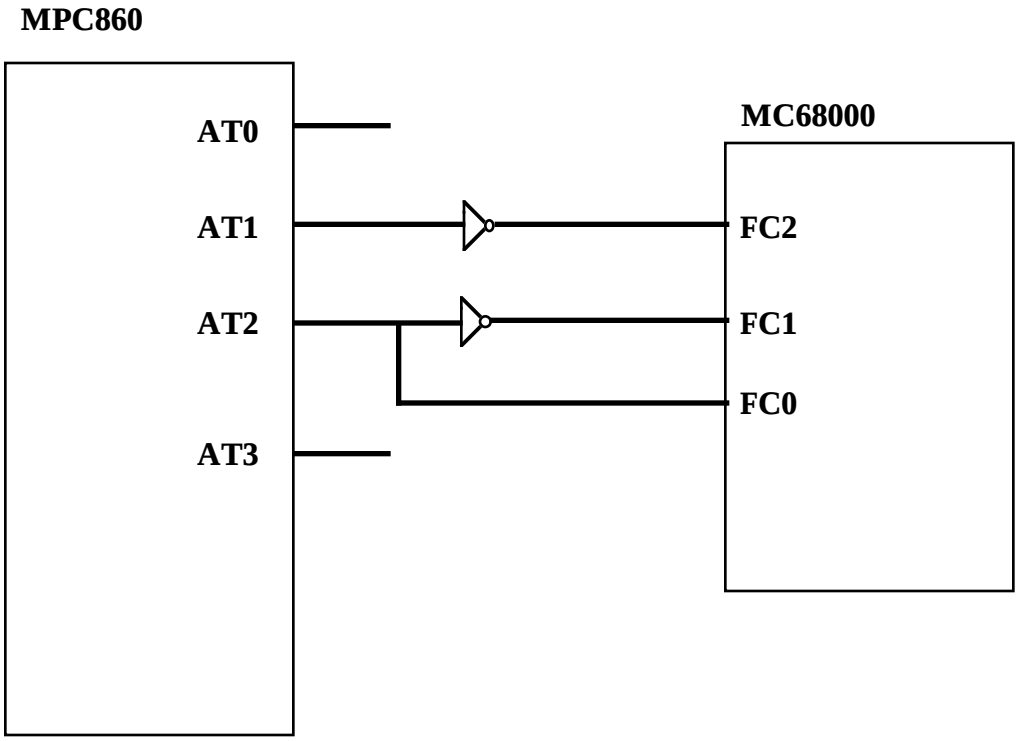


Figure 3. MPC860 Address Types Interfaced with Generic 68K Function Codes