Hitachi America, Ltd.

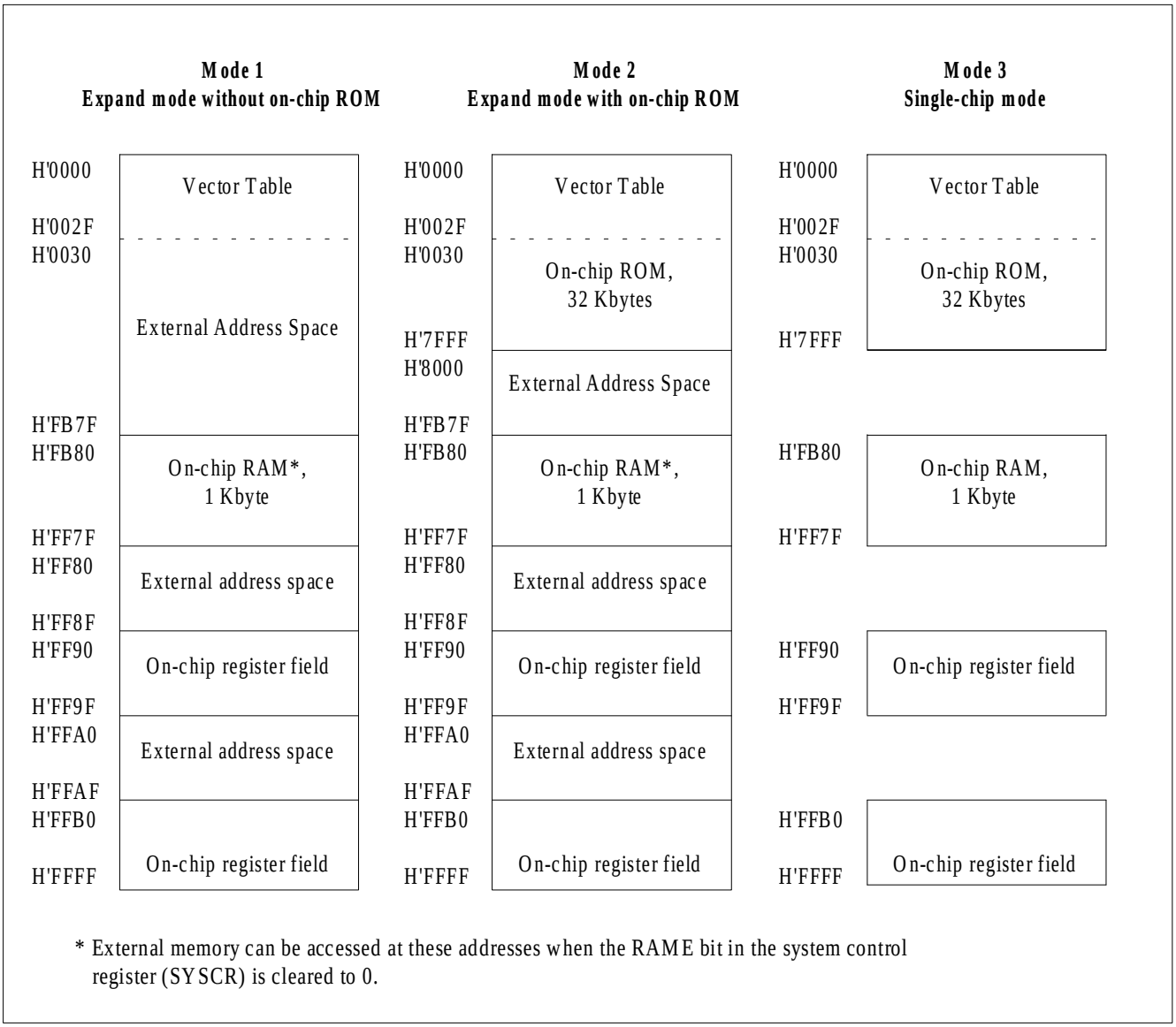
TN-0169

Application Engineering
TechNote

Sam Darwin

Interfacing the H8/325 Series to External Memory

The H8/325 Series of microcontrollers offers a variety of methods for managing memory space, based on the device's ability to operate in any one of three different modes. The mode determines how the memory space is allocated; whether it will be on-chip ROM and RAM, or if external memory space will be used instead. The modes are most easily understood by taking a look at the memory maps for each of the three different modes.



* External memory can be accessed at these addresses when the RAME bit in the system control register (SYSCR) is cleared to 0.

Figure 1 - H8/325 Address Space Map

Although these are the memory maps for the H8/325, they are quite similar to the memory maps of the other devices in the H8/300 family, with the main difference being memory size. Consult the user's manual for the specific details. To briefly summarize the information in Figure 1, Mode 1 is used if the program code is stored in external memory (external ROM). In this mode On-chip ROM is disabled, so external memory must be used. In Mode 2 On-chip ROM is enabled, so the program is stored on-chip, with either a mask ROM or a PROM option. Note that in both of these modes the RAM may be on-chip or off-chip depending on the setting of the RAME bit in the system control register. Mode 3 has both On-chip ROM and On-chip RAM.

To choose the mode setting, apply the appropriate inputs to the mode pins.

MD_1 pin	MD_0 pin	Mode
0	0	PROM programming mode. Do not use except during programming.
0	1	Mode 1 as shown above.
1	0	Mode 2.
1	1	Mode 3.

It is important that these pins are set correctly at the very beginning of processor operation, since the mode will determine where the first instruction will be coming from.

External Interface

In Modes 1 or 2, external interfacing is possible. The following pins are useful for this function.

Pin	Input or Output	Explanation
A_{15} to A_0	O	Address bus, used to address external memory.
D_7 to D_0	I/O	Data bus for interface with external memory.
Wait	I	Requests wait cycles during memory access.
RD	O	Read signal: Goes low to indicate the CPU is reading an external address.
WR	O	Write signal: Goes low to indicate CPU is writing an external address.
AS	O	Address strobe: Goes low to indicate valid address on bus.
IOS	O	I/O Select: Goes low when accessing H'FF00 - H'FFFF. Can be used as upper 8 bits of address bus.

Refer to the H8/325 manual for Bus Timing diagrams both for read and write. As with all memory accessing, certain timing parameters must be obeyed. There is also a diagram in the manual that illustrates the proper timing needed to insert wait states by pulling the wait signal low. This is useful for slower memories that require more access time either for reading or writing.

Example in Mode 1:

A simple way to configure memory in Mode 1 is to use the area from H'0000 to H'7FFF as external ROM, and the space from H'8000 to H'FF8F as external RAM. How can this be done? First, let's pick a couple of memory devices that will work effectively in this scenario. The HM62256 is a 32k x 8 bit SRAM with suitable access times to avoid wait states (if you are using other types of memory be sure to take a look at the access times to see if they are fast enough.) For the ROM we will use the HN27C256AG EPROM chip, which is also a 32k x 8 bit memory. Attach the H8/325 data and address buses to both memories as shown in Figure 2, with the exception of A_{15} which we will use to choose the memory we want to access. If A15 is 0, then the EPROM is enabled. If A15 is 1, then the external RAM must be enabled. Figure 2 illustrates the memory interface wiring needed.

Now that we have this basic system, what variations are possible? Other types of RAM may be used, but it is highly recommend to use 8 bit wide SRAMs, since the H8/325 family uses an 8-bit data bus and does not have the refresh signals required to run a DRAM system. Similarly, 8-bit wide ROM's should also be used. Ultimately, the H8/325 is able to interface with any number of different memories and other external devices as long as one main issue taken care of. External logic must observe the address lines in order to enable the proper device at the proper time. In the above example

the A_{15} line is used to pick between the two memories, but in more complex systems more address lines may be needed as well as more complex logic to make the decision. Most memories or other devices will have a "chip select" or "chip enable", and it is this that must be used to ensure that only one memory is being accessed at any given time.

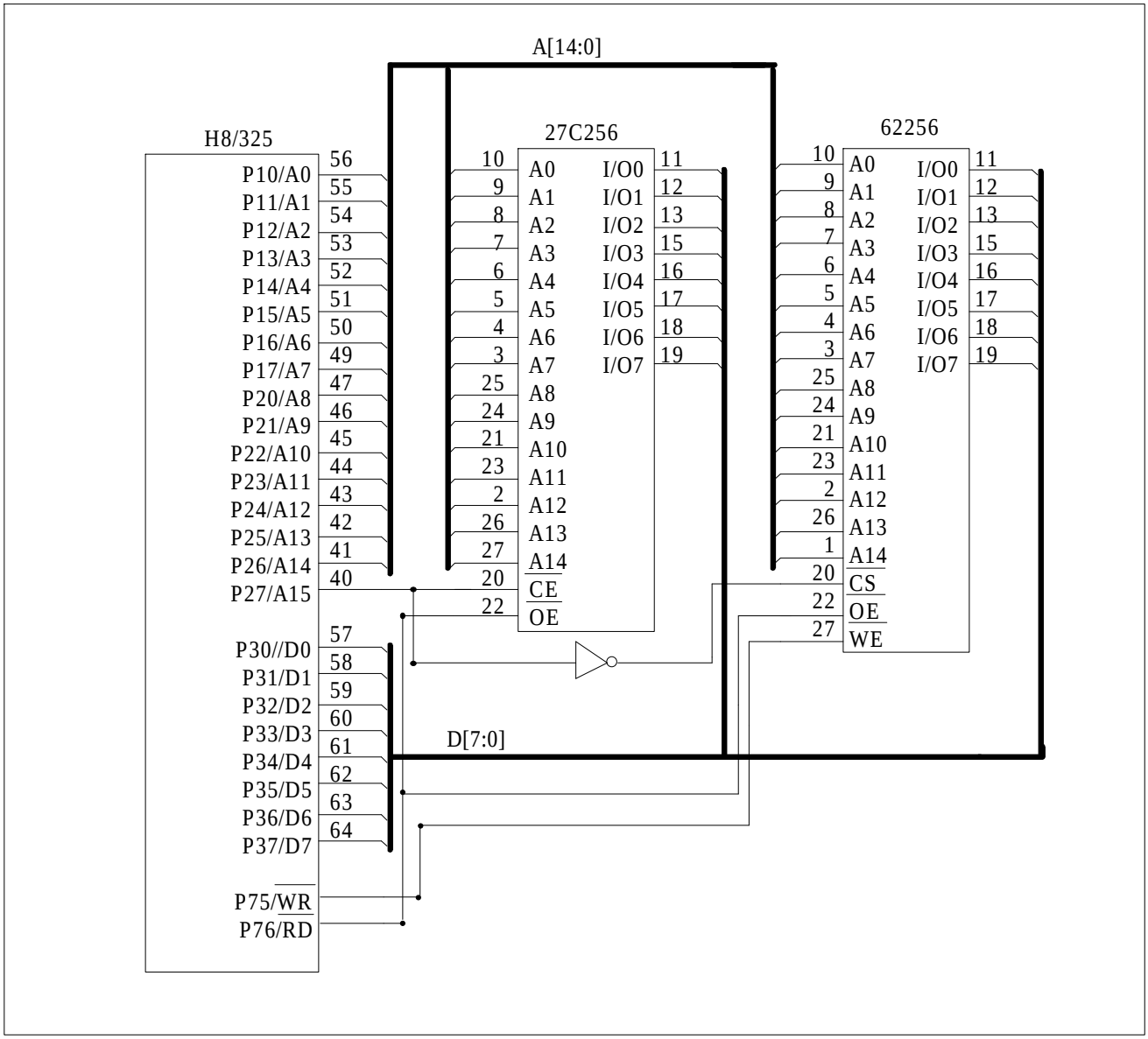


Figure 2 - H8/325 and Memory Schematic

Wait States

Look at the access time of the memory and decide if it is neccesary to add wait states. This is the case for slower memories that require more time than the H8/325 usually gives for either a read or a write. Each wait state lengthens the memory access timing by one clock cycle, but you should never add more wait states than neccessary because it slows down the processor operation. As an example, suppose that one wait state is required. The schematic show in Figure 3 illustrates how this might be accomplished, by initially causing a wait pulse when the Address Strobe line becomes low. The flip flop will then become high, but not before a wait state has been input to the processor. By varying this circuit slightly, 2 or 3 wait states could be added. The logic becomes even more complicated if different memories require different numbers of wait states. In the end there are many possible ways of implementing wait states, and we will leave it to the individual designer to find the best solution for his system.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

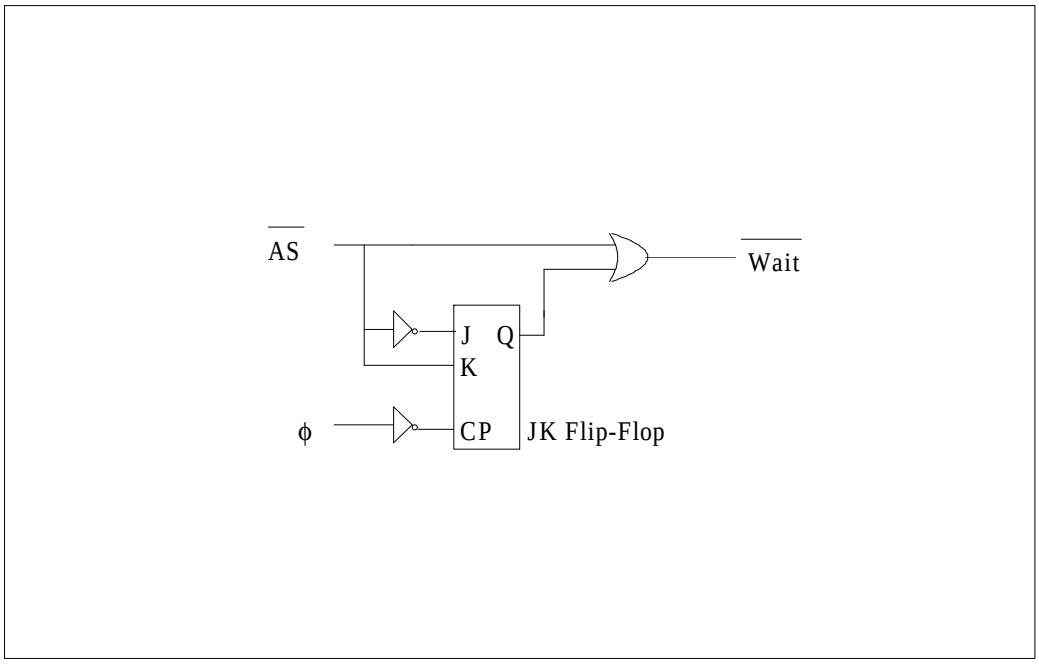


Figure 3 - Wait State Logic

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300