A Guide for Motorola HCS12 Expanded Mode Bus-Design and for Connecting an HCS12 Expanded Mode Target to a Nohau HCS12 Full-ICE

By: Doron Fael, Nohau

October 20, 2003

Introduction:

The Motorola HCS12 family of micro-controllers was designed primarily for operation in Single-Chip mode. In Single-Chip mode all the peripherals are found either internally in the HCS12 silicon, or can be connected to the HCS12 through one of its serial interfaces like the SPI, SCI or IIC. In practice over 95% of HCS12 based designs use Single-Chip mode.

Despite the above, an increasing number of HCS12 applications need to access an external memory from the HCS12 CPU, by using the HCS12 in Expanded mode. Since Motorola did not target the HCS12 family for extensive use in expanded modes, operation in this mode is not very easy to meet. In fact, the expanded bus of the HCS12 poses some significant obstacles from the hardware-design perspective. These obstacles need to be resolved in the board design stage (rather than in the board debugging stage) in order to allow proper operation of the HCS12 micro-controller with the expanded mode external components.

Chapter 1 of this guide explains the difficulties of designing reliable HCS12 expanded mode applications, and suggests how to resolve these difficulties. This chapter is general and can benefit all HCS12 Expanded Mode designers, including ones that don't use the Nohau HCS12 emulator or BDM.

Chapter 2 explains how to connect an expanded-mode target to a Nohau HCS12 full-Emulator for debugging. Some aspects need to be considered to allow such connection, and are discussed in details in this chapter.

Chapter 1: A Guide for Motorola HCS12 Expanded Mode Bus-Design

There are 4 main topics about HCS12 Expanded-Mode Bus-Design that need to be considered carefully in the design stage:

A. Compatibility of the HCS12 voltage levels when operated at 5V with the external memory voltage levels.
B. Very short multiplexed-address hold-time of 2nSEC after ECLK rise.
C. Very short Write-Data hold-time of 2nSEC after ECLK fall.
D. Signal Integrity and Transmission-Line effects.

These topics will be discussed in the following sections: 1.1 to 1.4.

1.1. Compatibility of the HCS12 voltage levels when operated at 5V with the external memory voltage levels

The HCS12 has CMOS input and output levels.
A logic-1 in an HCS12 input (Vih) is defined as 0.65*VDDX.
For VDDX of 5V, Vih is required to be 3.25V. Meaning – in order for the HCS12 to reliably read a logic 1 from an external memory on a data-bus input, it needs to read a voltage of 3.25V or higher in its input.

On the other hand, most memory devices (SRAM, FLASH etc.) have TTL input and output voltage levels. Specifically, most memory devices will only guarantee to drive Voh of 2.4V on their Data Bus outputs.

The 2.4V Voh that is guaranteed on the external memory data bus outputs is not compatible with the minimum 3.25V Vih, which is required in the HCS12 data bus inputs.

It would therefore be un-reliable design to connect the data bus of an external memory directly to the data bus of an HCS12 micro-controller, when the HCS12 is powered from 5V. Such direct connection should therefore be avoided.

In order to overcome the above problem one of the following two approaches can be used:

A. Use one of the recent HCS12 devices that has ability to be powered from either 5V or 3.3V, and power it from 3.3V.

At the time of writing this document (October 2003), the HCS12B, HCS12C, HCS12E and HCS12K families have the discussed ability to be powered from 3.3V or 5V.

When the HCS12 device is powered by VDDX of 3.3V:
Vih = 0.65 * VDDX = 0.65 * 3.3 = 2.145V

Meaning – in order for the HCS12 to reliably read a logic 1 from an external memory on a data-bus input, it needs to read a voltage of 2.145V or higher in its input.

On the other hand, the memory device guarantee Voh of 2.4V or higher in its data bus outputs which is compatible with the HCS12 data-bus inputs when the HCS12 is powered from 3.3V. The suggested method therefore constitutes a solution to the problem.

This method however has the following disadvantages:

- It cannot be used with all HCS12 devices (as not all HCS12 devices have the ability to be powered from 3.3V).
- It cannot be used for applications that require the HCS12 device to be powered by 5V.
- When the HCS12 device is powered from 3.3V it's expanded bus is only specified to operate up to a bus-speed of 16MHz. Higher bus-speeds above 16MHz, are not within the Motorola specification when the HCS12 is powered by 3.3V.

B. Use a bi-directional translation buffer on the data bus to translate the TTL data output from the memory-device to the CMOS input levels required at the data bus inputs of the HCS12 micro-controller.

Bi-directional transceivers such as 74VHCT245, 74AHCT245, 74ACT245, or some FCT245 are appropriate to be used as a "TTL to 5V-CMOS translation-buffer".

In this method, all the data bus information from/to the HCS12 and to/from the external memory is done through the translation buffer.

The disadvantages of this solution are:

- The Translation-Buffer inserts a delay of 5nSEC – 10nSEC in the data read and data write path. This delay may require the HCS12 external bus to be configured for 1 or more wait-states when the HCS12 is operated at high bus speed. Wait-states are configured by writing to the EXSTR[1..0] bits in the MISC register.
- The translation buffer has the ENA and DIR control inputs that need to be handled by additional logic. DIR can be typically connected to the HCS12 R/W signal (on PE2, and the ENA can be typically enabled by using some version of a fast-inverted ECLK.
- A fast decode-logic and a fast translation-buffer are recommended in order to avoid bus contention during the following address phase of the next bus cycle. Bus contention may occur after the ECLK fall, when the HCS12 is driving the address information of the following bus cycle, and the translation buffer still drives the previous external read data to the HCS12 address/data bus.

1.2. Very short multiplexed-address hold-time of 2nSEC after ECLK rise

The multiplexed address hold time after ECLK rise is very short (2nSEC is specified in the HCS12 data-sheets bus-timing).

In first glance, if not taking into consideration the discussed “short address hold time”, it seems a transparent latch from the 74xx373 family is the best candidate to latch the multiplexed address from the HCS12 address/data bus, so the address will be available to the external memory during the entire memory access time. The “most-natural” way to latch the address is to connect the input of an inverter to the HCS12 ECLK output, and run the output of this inverter to the LE (Latch-Enable) input of the 74xx373 latch.

However, the 74xx373 as well as most other address latches that are available in the market have active high LE (Latch-Enable) input. This means an inverter is needed between ECLK and the LE input of the address latch. Even if you use the fastest logic gate inverter, and the fastest address latch, you are not going to meet the specified 2nSEC of hold time, and will have a negative address hold time at the latch inputs. In such a case you will depend on the actual HCS12 address hold time being longer than the specified 2nSEC in order for the system to operate well. Needless to say – this would not be a very safe design method.

Solution A:

A partial solution to this problem is suggested by Motorola in their HCS12 external-mode reference design. An address register (74xx374) is used instead of an address latch. The advantage of this solution is that the ECLK can be connected directly to the CLK input of the 374 register to sample the address on the rising edge of ECLK (no inverter or any other logic is needed between the HCS12 and the address register). If you use one of the fastest address registers such as the 74ALVT16374, this may be a safe solution. However this solution too has a disadvantage: The address setup time to the memory is shortened in this method, since the address becomes available to the external memory only after the ECLK rise (after sampled by the '374 address register). It is therefore almost certain that this method will require setting one or more wait-state to be used for the external bus (in the MISC register EXSTR[1..0] bits) in order for the system to operate properly at bus speeds above 12MHz or so. For lower bus-speeds bellow 12MHz - zero wait-states may be sufficient.

Solution B

Use an address latch such as a 74xx373, and connect its LE (Latch Enable) input to the ECS or XCS outputs, which may be available on the HCS12 PK7 and PK6 outputs.

Note: ECS exist on most HCS12 parts on PK7. XCS exist only on some of the recent HCS12 parts such as the HCS12 “E” series.

The ECS or XCS act as “chip-selects” which become active for external memory accesses in certain address ranges. The timing of the ECS or XCS signals make them appropriate to be used to drive the LE input of an external latch. The ECS and XCS are by design guaranteed to be negated (become high) after every memory access. The limitation of this method is that the address will be latched appropriately only for accesses that make the used chip-select - ECS or XCS) become active (low).

ECS and XCS become high after ECLK fall, and become low again before ECLK rise (only for certain address ranges). This is the period that the address is valid on the HCS12 Port A and Port B pins in expanded modes. It should also be noted that this method will work only for address that make the ECS or XCS signals becomes active (low), as otherwise the address latch will not be latched during the data phase of the cycle.

The timing that is specified for ECS in the MC9S12D family data sheets for 5V operation is:
ECS becomes high (if not already high) at least 2nSEC after ECLK fall.
ECS becomes low not later than 16nSEC after ECLK fall (and before ECLK rise).
ECS is guaranteed to stay high for at least 8nSEC after ECLK fall and before ECLK rise.

The timing of the XCS does not exist in the Motorola documents. We can assume it has a similar timing like the ECS signal. ECS and XCS timing for 3.3V operation also do not exist in the Motorola documents. It is however safe to assume all the delays will be longer at 3.3V, as all HCS12 signal have higher slew-rate at 3.3V, and as external-mode operation is specified by Motorola to only 16MHz bus-speed at 3.3V. We can therefore assume the external bus timing is more relaxed when the HCS12 is powered from 3.3V, as compared to the timing when the HCS12 is powered from 5V.

XCS becomes active for every external address in the range 0000H - 7FFFH. This includes accesses to 4000H - 7FFFH when the ROMHM bit in the MISC register is set. XCS DOES NOT become active for external addresses in the range 8000H - BFFFH, even for PPAGE values 00H - 30H that should access external paged memory.

ECS becomes active for all external accesses to the range 8000H - BFFFH for all PPAGE values.

Deducing from the above:

A. XCS is a good candidate to be used as the LE signal for an address latch of memory at external addresses in the range 0000H - 7FFFH.
B. ECS is a good candidate to be used as the LE signal for an address latch of memory at external addresses in the range 8000H - BFFFH, when accessed by low PPAGE values of 00H - 2FH.
C. If external memory at both these address ranges is required, the ECS and XCS signals can be ANDed using a very fast AND gate (Pericom 74ST1G08 for example) and then applied to the address latch LE input.

In order to make ECS and XCS available on the external HCS12 bus, the EMK bit in the MODE register needs to be set to 1. This makes all Port K pins (including PK0 – PK5) act as paged address outputs, and chip-select outputs. None of the HCS12 PORTK pins can be used as a general-purpose I/O in this method.

1.3. Very short Write-Data hold-time of 2nSEC after ECLK fall

As specified in the HCS12 Data-Sheets in the "Expanded Bus Timing Characteristics", the HCS12 write-data is guaranteed to be held-valid for only 2nSEC after ECLK fall.

As a result, the decode logic cannot end the external memory write operation based on the ECLK fall. The reason is that the decode logic will likely have propagation delay of more than 2nSEC (more than the write data hold time after ECLK fall), and the external memory is therefore likely to be corrupted with invalid write data, as a result of the address information of the next memory access becoming available on the HCS12 Addr/Data bus a very short time after ECLK fall.

The solution for this problem is to use in the decode logic a delayed version of ECLK, which will end the external memory write operation delayed after ECLK rise, and before ECLK fall.

1.4. Signal Integrity and Transmission-Line effects

The HCS12 devices have relatively week and slow drivers on all the HCS12 outputs and I/Os, including on the HCS12 Address/Data and Control busses. This is in order for the HCS12 to preserve power when switching an output from one logic level to the other, and to supply good immunity to noise.

Every HCS12 output is recommended to drive not more than 10mA at full-drive and when operated at 5V, and not more than 5.5mA at low level and 4.5mA at high level at full-drive and when operated at 3.3V.

Every HCS12 output has a typical rise/fall time of 4nSEC with no load at 5V operation, and a typical 6nSEC rise/fall time with a 50pF load at 5V operation.

In Nohau's experience, these output characteristics and relatively high output impedance of the HCS12 drivers are not very good for expanded mode operation. This is noticed especially when attempting to operate at high bus speeds. When connecting the HCS12 outputs to complex address/data and control busses, impedance mismatch exists between the HCS12 output impedance and the impedance of the complex bus structures driven by these outputs. As a result much longer rise/fall times in the range of 10nSEC – 15nSEC can be experienced when the HCS12 is driving complex address, data and control busses.

In order to avoid these transmission-line effects and signal-integrity problems, it is recommended to minimize the length and the complexity of the HCS12 address, data and control busses.

Chapter 2: How to connect an expanded-mode target to a Nohau HCS12 full-Emulator for debugging

The full emulator has differences in its operation of emulating Expanded Mode operation, in compare with the operation of an HCS12 part. These differences are embedded into the Motorola HCS12 silicon and the emulator system. Following is a list and an explanation of these differences, and of how to connect an HCS12 expanded-mode target to the Nohau HCS12 emulator:

2.1. ECLK functional behavior

In the emulator, the ECLK signal (on PE4) will switch low and high in Expanded Mode for every bus access - both internal and external to the HCS12. This is to allow the emulator and the trace to observe both internal and external accesses of the CPU. For every bus cycle the address of the internal or external access will be driven to the target PORTA and PORTB pins when ECLK is low. If needed, the target can decode these addresses to determine when the access is to an internal location, and ignore it. Such decoding is needed to avoid inadvertent writing of internal resources, to external memory. If only external read-only memory is used, such address decoding will typically not be necessary.

2.2. ECLK Stretch for external memory for Wait-State insertion

The HCS12 CPU on the emulator is always running in Emulation Wide or Emulation Narrow mode. This is needed for the proper operation of the emulator. The correct operating condition (Normal Expanded Wide, Normal Single Chip, etc.) is reflected to the target using the Port Replacement Unit (PRU), which is part of the emulator system.

Since most HCS12 applications (>95%) operate the CPU in Single Chip mode, Motorola have made the ECLK stretch on HCS12 silicon behave in the following way in Emulation Wide and Emulation Narrow mode that the emulator uses:

All external addresses that normally would reside in the internal Flash are always configured to no ECLK stretch and no wait-states regardless of the settings of the MISC register EXSTR[1..0] bits (bits D3 and D2 of the MISC register). These include accesses to addresses 8000H - FFFFH always, and accesses to 4000H - 7FFFH when the MISC register ROMHM bit (bit D1) is cleared.

In other words, ECLK stretching on the full emulator is possible, and is following the MISC register EXSTR[1..0] bits only for the following memory accesses:

A. External accesses in the address range 0000H - 3FFFH, always.
B. External accesses in the address range 4000H - 7FFFH, when the ROMHM bit in the MISC register is set to 1.
C. Some HCS12 variants that normally have the internal Flash page 3DH accessible through address range 0000H – 3FFFH, also need to set ROMHM bit in the MISC register to 1 in order to configure more than 0 wait-states for these external accesses.

2.3. Delays between the Emulator Address/Data bus and the Target Address/Data bus

The Address/Data and Control buses that are reflected to the target on the PORTA, PORTB, PORTE and PORTK pins are not the same Address/Data and Control buses of the HCS12 CPU on the CPU-Module. There are buffers between these two busses in both directions: To the target in the case of Address, Control and Write Data, and from the target, in case of Read Data. In order to minimize the delay that these buffers create, the emulator uses the fastest buffers available, from the 74ALVT and 74LVT family.

In the case of Write-cycles, the address, control and the data is flowing from the CPU to the Target, and therefore the delays cancel each other, and the timing is similar to a real HCS12 part.

In the case of Read-cycles however, there are delays both in the direction from the CPU to the Target for the address and the control signals, and in the direction from the Target to the CPU for the read data. The designer should take into account a 9nSEC delay that is introduced by the emulator system in the timing of Tacca - Address Access Time - which is item 15 in the Expanded Bus Timing Characteristics of the HCS12 device Data Sheets (Address Valid to Read Data Valid).

2.4. ECLK Delays

The ECLK signal on the Target PE4 pin has slightly different timing from the ECLK signal of the HCS12 CPU on the emulator. In the default settings there is a buffer that introduces a delay of 4.5nSEC between the HCS12 CPU ECLK and the Target ECLK. In order to reduce this ECLK delay to only 2.5nSEC for Expanded mode operation, it is recommended to position of the ECLK jumper (located next to the 5 LEDs on the emulator motherboard) in the "FAST ECLK" position. This will improve the ECLK timing by 2 nSEC and will make the timing of the target memory system easier to meet.

2.4. Address Mapping

All the addresses that need to access the target memory need to be MAPPED to the target.
(If not mapped to the target, the emulator memory will be accessed instead of the target memory)
The mapping is done in the following way.
For example, in order to map address range 0400H - 0FFFH to the target do the following:

- In Seehau select Config -> Emulator ...
- Then select the "Map Config" tab.
- Click "ADD" and type in "0400" in the "Start address", and "0FFF" in the "End Address", then click OK.
- Make sure that the range 0400:0FFF shows up, and that the check-box is CHECKED! (If the check-box is not checked the emulator memory will be accessed and not the target memory)
- Now click "OK"

The address range 0400H - 0FFFH is now mapped to the target.

The mapping resolution is two-bytes (one aligned word). Each pair of two bytes can be mapped to the target memory or the emulation memory.

The following addresses can be effectively mapped to the target, and in the following settings:

A. When the "Emulate Rom in Expanded Mode" check-box in the Config -> Emulator -> Hdw Config is un-checked, and the emulator comes up after Reset in Normal Expanded mode:

   All the external addresses in the range 0000H - FFFFH may be mapped to the target as explained above.

B. When the "Emulate Rom in Expanded Mode" check-box in the Config -> Emulator -> Hdw Config is checked, or when the emulator comes up after Reset in Normal Single Chip Mode and then switched to expanded mode by writing to the MODE register:

   All the external addresses in the range 0000H - 3FFFH may be mapped to the target as explained above.
   All the external addresses in the range 4000H - 7FFFH may be mapped to the target as explained above, when the ROMHM bit of the MISC register is set to 1.
   The external addresses in the range 8000H - FFFFH MAY NOT be mapped to the target in this mode (see A above if this is important to enable).

2.5 Voltage Levels

When the emulator is used to emulate Expanded mode, all the Port A, B, E, and K signals that are recreated by the emulator's Port-Replacement-Unit (PRU) are driven with 3.3V drivers. The emulator is configured in this manner since 3.3V buffers are nowadays much faster than 5V buffers, and therefore allow higher speed operation of the external target memory interface. As a result, every Port A, B, E or K pin which is configured for the address / data or control bus function will be automatically configured by the emulator to drive 3.3V logic levels to the target.

All the Port A, B, E and K pins when configured to operate as general-purpose I/Os will have 5V CMOS output logic levels.

The described behavior is generally not a limitation for expanded mode operation, since the external memory devices usually have TTL levels, and are compatible with the 3.3V buffers used by the emulator. The 3.3V buffers used by the emulator are 5V tolerant and may be driven by 5V signals from the target memory system. The 3.3V buffers used by the emulator have much lower output impedance and faster rise/fall times as compared to the HCS12 outputs. This allows better rise/fall time than experienced on the HCS12 pins, and partially compensates for the delays introduced by the emulator buffers (discussed at section 2.3)