MOTOROLA
SEMICONDUCTOR
APPLICATION NOTE

Order this document by: AN1242/D

High Performance M68HC11 System Design Using The WSI PSD4XX and PSD5XX Families

by John Bodnar

INTRODUCTION

This application note covers conversion from a single-chip MC68HC711K4 microcontroller (MCU) system to a two chip MC68HC11K1 + PSD412A1 combination. It is not intended to be a comprehensive guide to using Motorola M68HC11 microcontrollers with WSI PSD4XX or PSD5XX microcontroller peripherals. These flexible devices provide a wide array of features, many of which cannot be adequately discussed within the context of this note. Designers with a more general interest in this topic should examine published material available from both Motorola and WSI. These documents are listed under **REFERENCES**.

GENERAL INFORMATION

M68HC11K MCUs are highly integrated derivatives of the MC68HC11F1, the first member of the M68HC11 family with a non-multiplexed address and data bus. Features common to M68HC11K devices include:

- M68HC11 CPU core capable of DC to 4 MHz operation
- Power saving STOP and WAIT modes
- 768 bytes of SRAM, with separate standby power input for battery backup
- Four programmable chip selects with clock stretching for expanded mode interfacing
- On-chip memory paging logic to allow expansion of the address space to 1 Mbyte
- 16-bit timer with programmable prescaler that includes 3 input capture (IC) channels, 4 output compare (OC) channels, and a single switchable IC or OC channel
- 8-bit pulse accumulator (PAC)
- Four 8-bit pulse width modulation (PWM) timer channels, pairs of which can be concatenated into two 16-bit channels
- Real-time interrupt circuit (RTI)
- Computer operating properly (COP) watchdog and clock monitor circuits
- Eight channel 8-bit analog-to-digital converter (ADC)
- Enhanced asynchronous non-return to zero serial communications interface (SCI)
- Enhanced synchronous serial peripheral interface (SPI)
- Maximum of 54 bits of bi-directional I/O available in single-chip mode of operation
- 8-bit fixed input only port

The MC68HC11K4 device provides 24 Kbytes of masked ROM, 768 bytes of SRAM, and 640 bytes of EEPROM. Cost-reduced versions of this device are available without EEPROM and/or masked ROM. The MC68HC711K4 is functionally equivalent to the MC68HC11K4 but has 24 Kbytes of one-time programmable or UV erasable EPROM instead of masked ROM. This programmable memory is typically used for prototyping, just-in-time inventory management, and applications requiring small production quantities or frequent code updates.

The MC68HC711K4 provides a great deal of flexibility for single-chip applications, but in some instances, it may be necessary to find an alternate solution that provides equivalent functionality. These situations may arise because:

- Pins used by the on-chip peripherals are also used to implement digital I/O ports, so use of peripherals can limit the available discrete digital I/O.
- Addition of new features may increase object code size beyond the 24 Kbytes provided by the internal EPROM. None of the M68HC11K devices provide more than 24 Kbytes of ROM or EPROM, and expanded operating mode uses 25 digital I/O pins for the address/data bus and read/write line.
- Reduced software maintenance costs due to source code development in a high-level language can initially be offset by greater object code size, causing a memory crunch and loss of I/O resources.
- Some applications require peripherals that are either not available on an M68HC11 derivative or not available with an SCI- or SPI-compatible interface.These devices usually have an address/data bus, and must be memory mapped, causing a loss of I/O resources.

Some of these problems may seem insurmountable without extensive hardware and/or software redesign, but there is a solution that offers both the flexibility of expanded operating mode and the I/O preservation of single-chip operating mode.

WSI PSD4XX and PSD5XX programmable system devices (PSDs) are peripherals with flexible bus interfaces that provide microcontroller system designers an integrated memory solution consisting of SRAM, EPROM, and programmable logic. PSDs can also provide up to 40 digital I/O lines to replace those occupied by the MCU address/data bus. PSD4XX and PSD5XX devices share the following features:

- bus access speeds of 90, 120, 150, and 200 nanoseconds
- 37 (PSD4XXA1), 59 (PSD4XXA2), or 61 (PSD5XX) PLD inputs
- 113 (PSD4XXA1), 126 (PSD4XXA2), or 140 (PSD5XX) PLD product terms
- 8 (PSD4XXA1), 24 (PSD4XXA2), or 30 (PSD5XX) registered macrocells
- 40 bi-directional digital I/O pins
- power management unit (PMU)
- 32K x 8 (PSDX11), 64K x 8 (PSDX12), or 128K x 8 (PSDX13) of EPROM
- 2K x 8 of SRAM

The PSD5XX family builds upon the PSD4XX family by adding a peripheral module that contains four 16-bit counters/timers, a watchdog timer, and an eight level interrupt controller. On all PSDs, some of the programmable logic is used to map the different memory blocks and control registers for the I/O ports, the PMU, and (on PSD5XX devices) peripheral control registers. Usually, sufficient programmable logic remains after memory decoding to implement chip-select signals for external memory mapped peripherals, other bus control signals, and even state machines to perform useful peripheral functions.

The discussion which follows covers the process of converting a hypothetical single-chip M68HC11 application to an equivalent or enhanced “two-chip solution” consisting of a non-multiplexed bus M68HC11 and a PSD4XX or PSD5XX device. Please refer to the **DEVICE REFERENCE TABLES**, at the end of this note, for a list of compatible devices. **Table 4** shows suitable non-multiplexed bus M68HC11 devices. **Table 5** shows PSD4XX and PSD5XX devices.

THE PROBLEM

Figure 1 shows a single-chip MC68HC711K4 application that makes extensive use of MCU on-chip peripheral and memory resources. An ASCII terminal interface that facilitates user interaction is an important feature of this design. However, the customer has requested that the next generation product be substantially smaller. This can best be achieved by eliminating the ASCII terminal.

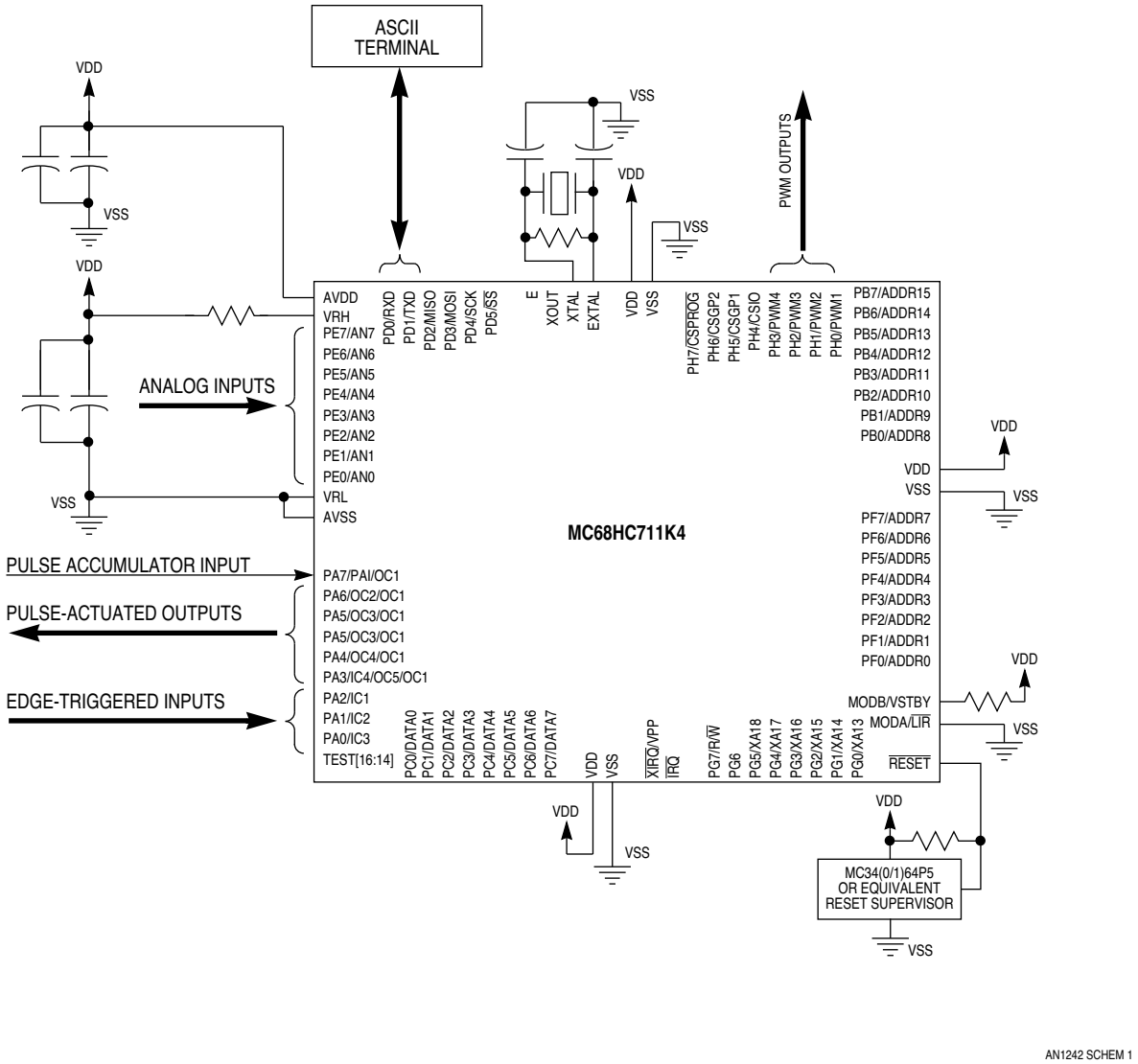

Figure 1 Existing Single-Chip MC68HC711K4 Application

The redesign promising the greatest size reduction integrates a large LCD panel and keyboard with an M68HC11-based control unit. These changes meet customer requirements for a more compact, more tightly integrated control solution, but as **Figure 2** indicates, the keyboard and LCD interfaces could exhaust the digital I/O resources of the MC68HC711K4.

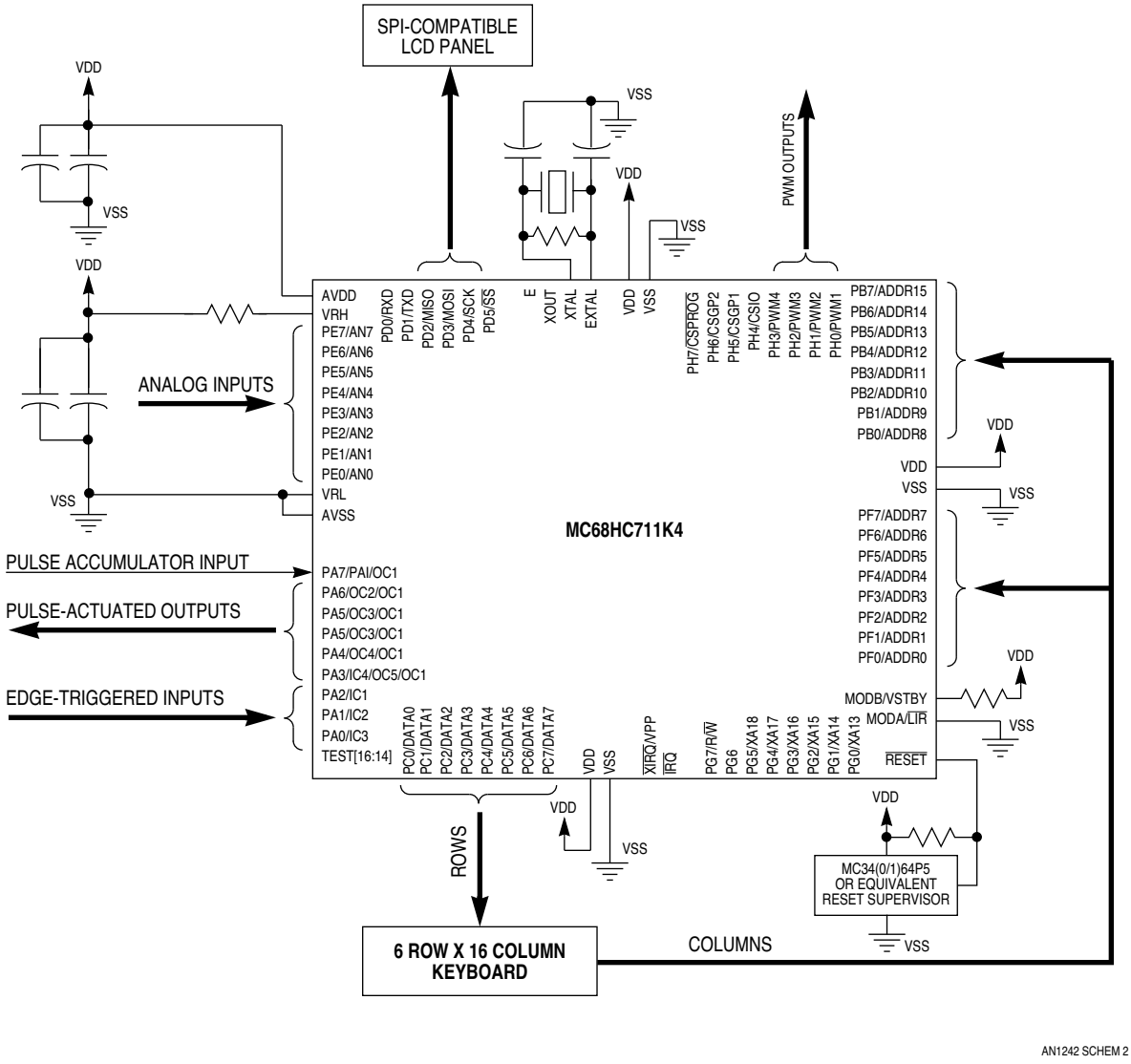

Figure 2 Next Generation MC68HC711K4 Application With Keyboard and LCD

To complicate matters further, these changes make greater demands of the MCU firmware. As shown in **Figure 3**, application control code and constant tables fit neatly into the 24 Kbytes of EPROM on the MC68HC711K4. The ASCII terminal connection to the SCI reduces the user interface to simple character I/O functions, but addition of a parallel input keyboard and a large LCD panel requires supplemental firmware support that increases permanent storage requirements to more than 24 Kbytes.

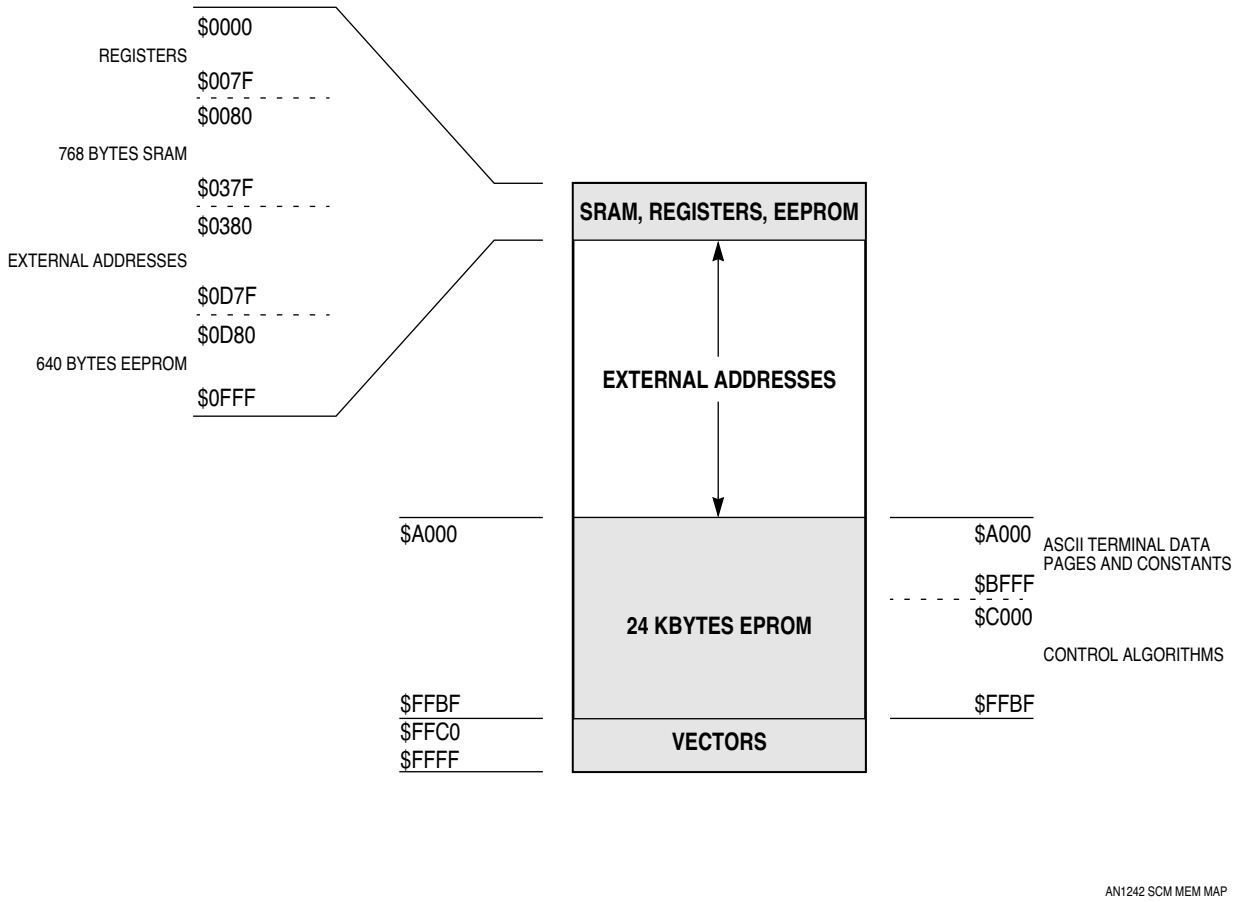

Figure 3 Initial Single-Chip Mode Memory Map

Sophisticated control algorithms are required to support the increased functionality of the keyboard and LCD panel. To speed code development, reduce the cost of support, and provide for future enhancements, the firmware for the next generation product will be ported from M68HC11 assembly language to C. While high-level languages simplify the development of complex control applications, they do so at the expense of greater storage requirements. The reduction in object code size achieved by hand-tuning assembly language programs begins to disappear as application functionality and complexity increase.

One possible method of providing for increased storage demands would be to use the flexible memory expansion capabilities of an M68HC11K device. This would lead to the simple expanded memory map shown in **Figure 4**. Estimates indicate that the control algorithms will require 32 Kbytes of EPROM and that the LCD data tables will require an additional 32 Kbytes. A 16 Kbyte memory paging window can be used to access the LCD data and still provide 12 Kbytes of contiguous address space for any other memory mapped peripherals that may be needed.

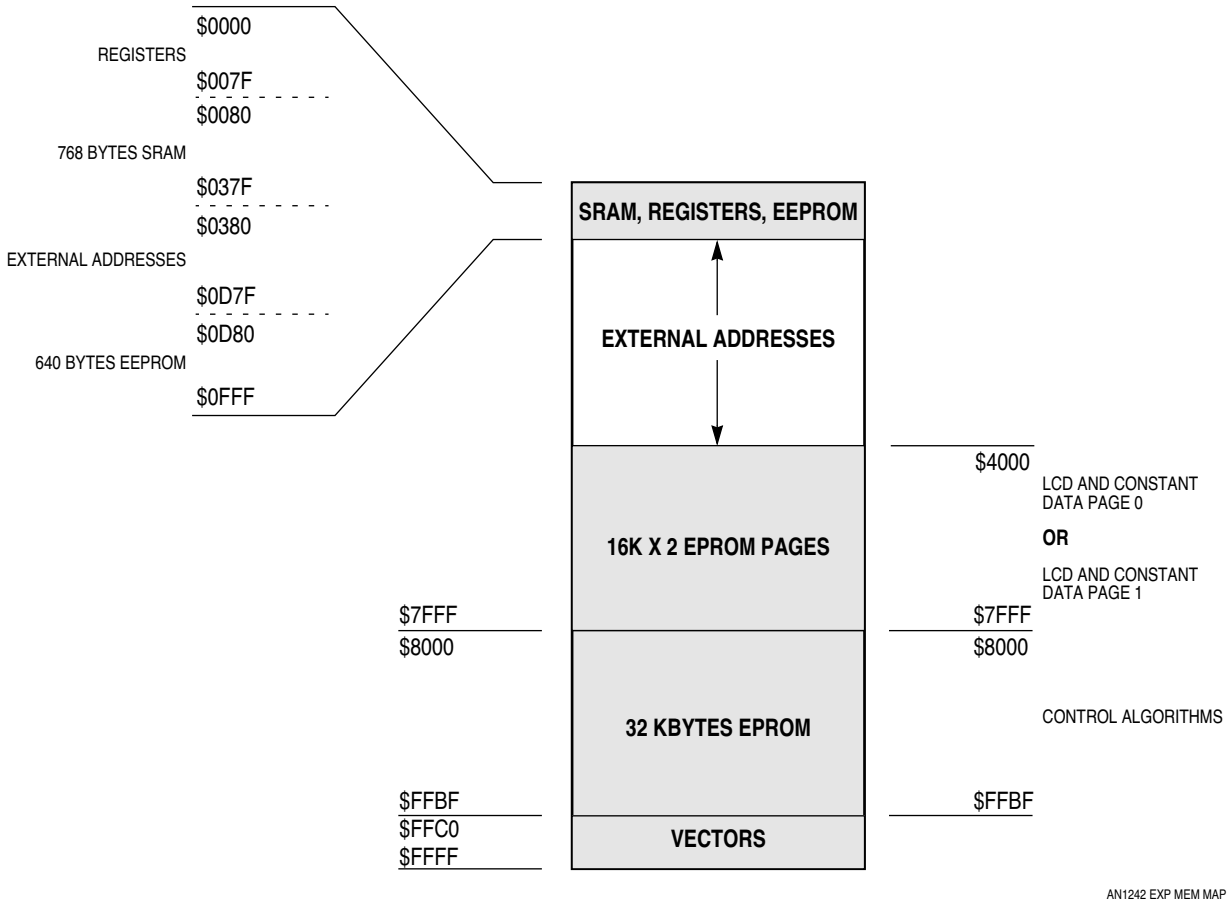

Figure 4 Proposed Memory Map Expansion

Unfortunately, access to the MC68HC711K4 address and data buses results in the complete loss of I/O Ports B, C, and F, as well as bit 7 of Port G. In addition, other Port G bits would be used as expansion address lines and some Port H bits may be used as chip selects. These lost I/O pins can be rebuilt with simple latches or more complex peripherals at the risk of decreased flexibility and more complicated circuit design and debugging.

THE SOLUTION

Before proceeding with a design solution based on a WSI PSD4XX or PSD5XX device, it is instructive to review the problem as it now stands.

- The existing MC68HC711K4-based system makes extensive use of the integrated MCU peripheral resources. In particular, the SCI connects to an ASCII terminal that simplifies interactive user control.
- Size reductions specified for the next generation of this product are best achieved by replacing the ASCII terminal with a keyboard and LCD panel that are integrated with the control unit.
- The digital I/O requirements for the keyboard and LCD interfaces could exhaust the MC68HC711K4 I/O resources.
- Application storage demands increase for two reasons. The keyboard and LCD panel require additional interface code, and the firmware is to be ported from assembly language to C.
- To support the proposed expansion of storage capacity from 24 Kbytes to 64 Kbytes, the MCU must use expanded operating mode rather than single-chip operating mode.
- Expanded mode operation requires at least 25 pins for the non-multiplexed address/data bus and the read/write line. These pins are currently used for digital I/O. Other digital I/O pins must be used to support chip selects and memory expansion beyond 64 Kbytes.

It appears that the proposed system would require a 32 Kbyte EPROM for control routines, another 32 Kbyte EPROM for LCD data tables, several latches to rebuild lost I/O ports, and some programmable logic to map all of these devices into the MC68HC711K4 address space. This design clearly jeopardizes cost savings achieved by the existing implementation and future high-level language software development.

Some of the savings can be restored by switching MCUs. The 24 Kbyte EPROM on the MC68HC711K4 is not needed for the new design, so either the ROM-less MC68HC11K1 or the ROM-and-EEPROM-less MC68HC11K0 could be used. These devices retain the specialized peripherals available on the MC68HC(7)11K4, and are ideal for expanded mode applications where the I/O pins used by the bus interface are not required or are otherwise rebuilt.

Both cost reduction and increased flexibility can be achieved by using a WSI PSD4XX or PSD5XX programmable system device in place of the memory and logic components that would otherwise be needed to realize this design.

As shown in **Table 4** and **Table 5**, a PSD412A1 can easily provide the required additional memory, I/O, and logic resources. If subsequent specifications dictate increased memory, logic, or even peripheral functionality, other members of the PSD4XX and PSD5XX families could be used, while maintaining close compatibility with the PSD412A1.

Table 1 compares the memory, I/O, and logic resources of both the initial MC68HC711K4 system and the proposed MC68HC11K(0/1) + PSD412A1 system

.

Table 1 M68HC11 Single-Chip versus M68HC11 + PSD4XX Resource Comparison

	MC68HC711K4	**MC68HC11K(0/1) + PSD412A1**
EPROM	24 Kbytes	16 Kbytes + 16 Kbytes + 32 Kbytes = 64 Kbytes
SRAM	768 bytes	768 + 2048 = 2816 bytes
EEPROM	640 bytes	640 bytes/0 bytes
Available Bi-directional I/O	54 lines	61 lines
PLD input terms	None	61
PLD product terms	None	113
Registered Macro Cells	None	8

In essence, the combination of a non-multiplexed bus M68HC11 MCU and a PSD4XX or PSD5XX device restores much of the functionality of a single-chip system. While not providing the ultimate size and power consumption features of such a design, the increased flexibility of this pairing and the freedom it provides to system designers is a competitive advantage.

THE CONVERSION PROCESS

Converting a single-chip M68HC11 application to a two chip system consisting of a non-multiplexed bus M68HC11 and a PSD4XX or PSD5XX is a five step process:

1. Assess the memory, I/O, and logic requirements of the combined system
2. Select the appropriate M68HC11 and PSD combination.
3. Produce the two chip system memory map.
4. Determine which PSD I/O ports replace M68HC11 I/O ports used for expanded mode operation.
5. Generate a schematic for the combined system.

1. Assess The Memory, I/O, And Logic Requirements Of The Combined System

This step has already been discussed. Key determinations to be made in this step include:

- How much I/O is required for the combined application?
  - — Consider single-chip usage and any additional I/O that will be necessary for current and/or future product enhancements.
- How much memory is required for the combined application?
  - — Consider potential firmware enhancements and the possibility of source code migration from assembly language to a high-level language like C or a visual application builder.
  - — Also consider additional RAM requirements. PSD4XX and PSD5XX devices provide 2K x 8 of SRAM that can be powered from backup batteries, and future derivatives may eliminate the SRAM to reduce cost. If even more RAM is necessary, the PSD can provide the decode logic needed to memory map larger devices.
- How much logic will be required?
  - — Any conversion to a two chip solution will use at least some of the PSD decode logic for memory, I/O port, and control register mapping. If the existing single-chip system uses PALs or 74HC family logic, consider using the PSD to replace as much of this as possible. The lower chip count reduces cost and use of the PSDsilos III™ simulation software can reduce system debug time.

2. Select The Appropriate M68HC11 And PSD Combination

Choose the M68HC11 and PSD combination carefully.

- When compatibility between the single-chip M68HC11 system and its PSD-based expanded mode counterpart is essential, use a ROM-less version of the single-chip MCU. In the example application, an MC68HC11K1 or an MC68HC11K0 can be paired with the PSD412A1 to replace the MC68HC711K4.
- In applications where maximum compatibility is not required, carefully selecting the M68HC11 MCU and PSD can realize considerable cost savings.
  - — If the M68HC11 device is used because it has a large EPROM array, consider replacing it with a smaller ROM-less derivative. The PSD can be chosen to maximize available EPROM and I/O.
  - — If the M68HC11 device is used to provide large amounts of I/O, choose the nearest equivalent ROM-less version and a PSD that will maximize available I/O.
  - — If the M68HC11 device is used for high-speed execution, consider using a smaller ROM-less derivative capable of the same performance. The PSD can be chosen to maximize available EPROM and I/O.
  - — If the M68HC11 device is used because it has a specific on-chip peripheral complement, choose the nearest equivalent ROM-less version and PSD that approximate the functionality of the single-chip device.
- Selection of an appropriate PSD is relatively straightforward. The device must meet the memory, I/O, and logic requirements determined in Step 1. If necessary, the MCU can be chosen to augment PSD resources, such as I/O and logic used for chip selects.

3. Produce The Two-Chip System Memory Map

This step is best explained by continuing with the example application. First, examine the memory map of the M68HC11 derivative to be used and locate areas not occupied by internal memory resources. These openings in the 16-bit address space are available for memory mapping external devices. For MC68HC11K(0/1) devices, the following ranges are externally addressable.

$0380 to $0D7F

$1000 to $FFFF ($2000 to $FFFF if CSIO is used)

A 60 Kbyte block of space is available from $1000 to $FFFF in expanded operating mode. However, if the I/O chip select (CSIO) function implemented in M68HC11K devices is used, this area is reduced to 56 Kbytes from $2000 to $FFFF.

Allocating space to CSIO allows use of a memory-mapped display controller instead of a display controller with a serial or a parallel interface. A number of manufacturers provide a complete LCD solution that includes an intelligent display controller. The controller can be connected directly to a microcontroller address/data bus if slow access times can be managed. The CSIO signal is ideal for this purpose because it can be stretched by up to three E clock cycles. CSIO requires the fixed 4 Kbyte block of addresses from $1000 to $1FFF in order to operate.

Compile the memory map for the two chip system by listing the following address ranges:

- M68HC11 SRAM
- M68HC11 register block
- M68HC11 EEPROM, if used
- M68HC11 fixed chip select address ranges, if used
- 256 byte PSD register block
- PSD SRAM, if used
- PSD EPROM blocks

Figure 5 shows the combined memory map for the example application.

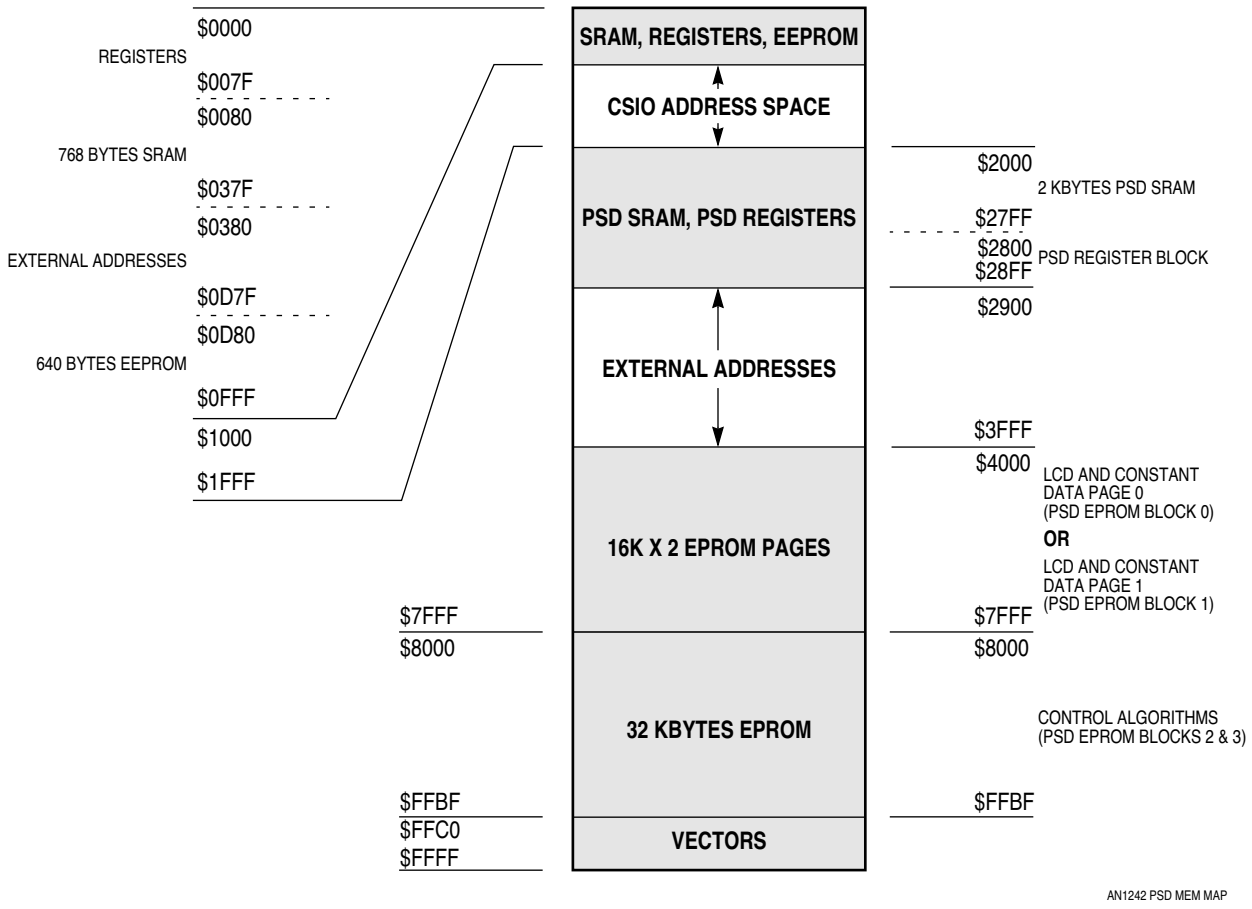

Figure 5 Combined MC68HC11K1 + PSD412A1 Memory Map

The ultimate purpose of this memory map is to guide development of a PSDabel™ file. PSDabel™ is one component of WSI's comprehensive PSDsoft™ design package that also includes PSDcontrol™ (configuration, compilation, de-compilation, fitting, address translation, hex data file conversion, and device programming) and PSDsilos III™ (Verilog-based device simulation). PSDabel™ is based on Data I/O Corporation's ABEL Hardware Description Language. It is used to describe the logical operation of the PSD4XX and PSD5XX decode ZPLD (DPLD) and general purpose ZPLD (GPLD).

A listing of the PSDabel™ file used to implement the memory map shown in **Figure 5** follows. The included comments provide a basic understanding of how a PSDabel™ file is constructed. Refer to the *PSDabel™ Manual* for further documentation and a tutorial.

```
module K4_TO_PSD_CONVERSION
title 'MC68HC711K4 to MC68HC11K1 + PSD412A1 Conversion'

"The following section lists the input signals.

"First come the address lines using their reserved names. Note that only those signals
"listed are routed to the DPLD.

a15,a14,a13,a12,a11,a10,a9,a8,a1,a0 pin;

"Next come the general purpose inputs used for the paging scheme. Uncomment the lines
"implementing the desired paging. For this application, the PSD page register will be
used
"because it requires no additional I/O pins. The K1 memory expansion address lines may be
"used if additional address bits or the page register inputs are required for specific
"decoding purposes. Use of the page register will be discussed later.

pgr3,pgr2,pgr1,pgr0 pin;  "These are the 4 input bits of the PSD page register.

xapage pin 20;   "This is XA14 from the MC68HC11K1 and is used to select one of the 16K LCD
                  "data table pages.

"The M68HC11 non-multiplexed bus control signals are specified here.

rd_wr,e pin 29,41;   "M68HC11 R/W* and E specified here as PSD pins 29 (WR) and 41 (RD).

"Now the DPLD chip select outputs are listed.

"CSIOP is the chip select for the PSD register block.
"RS0 is the chip select for the 2K PSD SRAM.
"ES[0:3] are the chip selects for PSD EPROM blocks 0, 1, 2, and 3.

csiop,rs0,es0,es1,es2,es3 node;

"Signal definitions and groupings now follow.

X = .x.;  "This is how a don't care term is specified.

"This definition groups together the CPU address lines.

CPUaddress = [a15,a14,a13,a12,a11,a10,a9,a8,X,X,X,X,X,X,a1,a0];

"This definition groups together the page register bits.

PAGE = [pgr3,pgr2,pgr1,pgr0];

"DPLD Chip Select Equations.

"This maps the PSD register block from $2800 to $28FF.

csiop = (CPUaddress >= ^h2800) & (CPUaddress <= ^h28FF);

"This maps the PSD 2K SRAM from $2000 to $27FF.

rs0 = (CPUaddress >= ^h2000) & (CPUaddress <= ^h27FF);

"This maps 16K PSD EPROM block 3 from $C000 to $FFFF.

es3 = (CPUaddress >= ^hC000) & (CPUaddress <= ^hFFFF);

"This maps 16K PSD EPROM block 2 from $8000 to $BFFF.

es2 = (CPUaddress >= ^h8000) & (CPUaddress <= ^hBFFF);

"This maps 16K PSD EPROM block 1 from $4000 to $7FFF when XA14 is logic one, i.e. this
is LCD
"data table page 1. Use this equation when the K1 memory expansion is used in place of the
"PSD page register.

"es1 = xapage & (CPUaddress >= ^h4000) & (CPUaddress <= ^h7FFF);

"This maps 16K PSD EPROM block 1 from $4000 to $7FFF when PAGE = $1, i.e. this is LCD
"data table page 1. Do not use this equation if the K1 memory expansion is being used.

es1 = (PAGE == ^h1) & (CPUaddress >= ^h4000) & (CPUaddress <= ^h7FFF);

"This maps 16K PSD EPROM block 0 from $4000 to $7FFF when XA14 is logic zero, i.e. this
is LCD
"data table page 0. Use this equation when the K1 memory expansion is used in place of the
"PSD page register.

"es0 = !xapage & (CPUaddress >= ^h4000) & (CPUaddress <= ^h7FFF);

"This maps 16K PSD EPROM block 0 from $4000 to $7FFF when PAGE = $0, i.e. this is LCD
"data table page 0. Do not use this equation if the K1 memory expansion is being used.

es0 = (PAGE == ^h0) & (CPUaddress >= ^h4000) & (CPUaddress <= ^h7FFF);

end K4_TO_PSD_CONVERSION
```

4. Determine Which PSD I/O Ports Replace M68HC11 I/O Ports

PSD4XX and PSD5XX devices have five 8-bit I/O ports, labeled A, B, C, D, and E. When used with a non-multiplexed bus M68HC11, Port C becomes the 8-bit data bus. Of the available 32 bits of general purpose I/O, 24 are used to rebuild M68HC11 Ports B, C, and F (the address/data bus), and the remaining 8 can be used to rebuild Port G bit 7 (the R/$\overline{W}$ line) and other Port G or Port H I/O pins used for expansion address lines or chip selects.

To modify existing single-chip M68HC11 software to take advantage of PSD I/O ports, simply substitute PSD register addresses for HC11 register addresses. A typical M68HC11 I/O port has both a data direction register and a port data register. Every PSD I/O port has a control register that determines port function, a data direction register, a data in register, and a data out register. Some PSD I/O ports also have registers to enable open drain operation, to determine if a pin is used as a PLD signal or I/O bit, and to read the outputs of the GPLD.

A good way to view the port relationships between a PSD and an M68HC11 is to construct a table that lists each port and its associated registers. On one side of the table, list the M68HC11 I/O port and its registers, and on the other side, list the equivalent PSD I/O port and its registers. Use this table as a guide when modifying single-chip firmware to support the two chip M68HC11/PSD system. **Table 2** is an I/O mapping table for the example application. Remember that CSIOP is mapped from $2800 to $28FF.

Table 2 M68HC11 to PSD I/O Conversion Table

MC68HC711K4			PSD412A1		
Port B	DDRB	$0002	Port B	PB_DDR	$2807
	PORTB	$0004		PB_INDATA	$2801
				PB_OUTDATA	$2805
				PB_CONTROL	$2803
				PB_PLD_IO	$280B
				PB_MAC_OUT	$280D
Port C	DDRC	$0007	Port A	PA_DDR	$2806
	PORTC	$0006		PA_INDATA	$2800
				PA_OUTDATA	$2804
				PA_CONTROL	$2802
				PA_PLD_IO	$280A
				PA_MAC_OUT	$280C
Port F	DDRF	$0003	Port E	PE_DDR	$2826
	PORTF	$0005		PE_INDATA	$2820
				PE_OUTDATA	$2824
				PE_CONTROL	$2822
				PE_PLD_IO	$282A
				PE_MAC_OUT	$282C
Occupied Port G/H I/O	DDRG/H	$007F or $007D	Port D	PD_DDR	$2817
	PORTG/H	$007E or $007D		PD_INDATA	$2811
				PD_OUTDATA	$2815
				PD_CONTROL	$2813
				PD_OPN_DRN	$2819

As **Table 2** indicates, the location of each PSD register is specified as an 8-bit offset from the CSIOP base address specified in the PSDabel™ file. The PSD4XX and PSD5XX documentation lists these 8-bit offsets.

A few small differences in I/O functionality should be noted:

- I/O ports on some M68HC11 devices have assignable pull-up resistors. For example, the PPAR register at $002C on M68HC11K MCUs can enable pull-up devices on Ports G and H in all modes and on Ports B and F only in single-chip mode. This feature is not available on PSD4XX or PSD5XX devices, so external pull-ups may be needed.
- I/O Ports B, C, and F on M68HC11K MCUs do not have any sort of control or alternate function registers, although Port C can be placed in open drain mode with the CWOM bit in the OPT2 register at $0038. If this functionality must be maintained, replace Port C on the MC68HC711K4 with PSD4XX or PSD5XX Port D. The open drain control register (PD_OPN_DRN in the table above) allows each PSD Port D I/O pin to be configured for normal or open drain mode.
- The PGAR register at $002D is used to enable the memory expansion address lines associated with Port G bits 0 to 5. Setting bits in this register to 1 overrides the Port G I/O functions and enables the associated XA lines. This register is set to $00 after reset.
- Chip select control registers CSCTL, GPCS1A, and GPCS2A, located respectively at $005B, $005C, and $005E, override the I/O functions of Port H bits 4 to 7. In expanded operating mode, GPCS1A and GPCS2A are set to $00 after reset, thus disabling general purpose chip selects 1 and 2 (CSGP1 and CSGP2). CSCTL will be set to $04 after reset, leaving the I/O chip select (CSIO) disabled and the program chip select ($\overline{\text{CSPROG}}$) enabled. Write CSCTL to $00 to disable $\overline{\text{CSPROG}}$ and make the PH7/$\overline{\text{CSPROG}}$ pin available for I/O. $\overline{\text{CSPROG}}$ is not required for interfacing to the PSD, although it can be used in conjunction with the PSD power management unit (PMU) to reduce power consumption.

The code examples that follow demonstrate how the PSD I/O ports are accessed in comparison with M68HC11 I/O ports. Access to the other PSD control registers is achieved in the same straightforward fashion. Please refer to PSD4XX and PSD5XX documentation for more information.

Single-Chip MC68HC711K4

```
*
* port B, C, and F I/O
*
REGBASE equ   $0000
DDRB    equ   $02
PORTB   equ   $04
DDRC    equ   $07
PORTC   equ   $06
DDRF    equ   $03
PORTF   equ   $05
*
* read port B[7:0]
*
        clr   REGBASE + DDRB
        ldaa  REGBASE + PORTB
*
* write pattern to port C[7:0]
*
        ldaa  #$FF
        staa  REGBASE + DDRC
        ldaa  #$55
        staa  REGBASE + PORTC
*
* configure PF[3:0] for inputs,
* PF[7:4] for outputs, poll until PF0
* is set to 1, then write pattern to
* PF[7:4].
*
        ldx   #REGBASE
        ldaa  #$F0
        staa  DDRF,X
POLLPF0 brclr PORTF,X,$01,POLLPF0
        bset  PORTF,X,$A0
```

MC68HC11K(0/1) + PSD412A1

```
*
* port B, A, and E I/O
*
REGBASE    equ   $2800
PB_DDR     equ   $07
PB_INDATA  equ   $01
PB_OUTDATA equ   $05
PB_CONTROL equ   $03
PA_DDR     equ   $06
PA_INDATA  equ   $00
PA_OUTDATA equ   $04
PA_CONTROL equ   $02
PE_DDR     equ   $26
PE_INDATA  equ   $20
PE_OUTDATA equ   $24
PE_CONTROL equ   $02
*
* make ports B, A, and E exclusively
* available for I/O
*
           ldaa  #$FF
           staa  REGBASE + PB_CONTROL
           staa  REGBASE + PA_CONTROL
           staa  REGBASE + PE_CONTROL
*
* read port B[7:0]
*
           clr   REGBASE + PB_DDR
           ldaa  REGBASE + PB_INDATA
*
* write pattern to port A[7:0]
*
           ldaa  #$FF
           staa  REGBASE + PA_DDR
           ldaa  #$55
           staa  REGBASE + PA_OUTDATA
*
* configure PE[3:0] for inputs,
* PE[7:4] for outputs, poll until PE0
* is set to 1, then write pattern to
* PE[7:4].
*
           ldx   #REGBASE
           ldaa  #$F0
           staa  PE_DDR,X
POLLPE0    brclr PE_INDATA,X,$01,POLLPE0
           bset  PE_OUTDATA,X,$A0
```

5. Generate A Schematic For The Combined System

Table 3 shows the connections between a non-multiplexed bus M68HC11 and a PSD4XX or PSD5XX

Table 3 M68HC11 to PSD Connections

M68HC11	PSD4XX or PSD5XX
ADDR[15:0]	ADIO[15:0]
DATA[7:0]	PC[7:0]
E	RD
R/$\overline{W}$	WR

An expansion address line (XA14) could be connected to one of the PSD Port A inputs, and used to select the 16 Kbyte LCD table EPROM pages. In the example application, however, it is easier to use the PSD page register. The four page register bits (PG[3:0]) can be used as inputs to the DPLD. In the example PSDabel™ listing, the ES0 and ES1 EPROM chip selects are decoded when the page register value is $0 or $1 and the CPU address is between $4000 and $7FFF.

The page register is accessed as follows.

```
REGBASE   equ   $2800
PSD_PAGE  equ   $E0
PAGE0     equ   $00
PAGE1     equ   $01
LCD_LINE1 equ   $4000
LINE_LEN  equ   $F0
*
* select EPROM page 0/LCD table 0
*
          ldaa  #PAGE0
          staa  REGBASE + PSD_PAGE
*
* read data from selected page
*
          ldx   #LCD_LINE1
          ldab  #LINE_LEN
SEND_L1   ldaa  0,X
          jsr   SEND_DATA
          inx
          decb
          bne   SEND_L1
          .
          . etc.
          .
```

The page register bits are available as inputs to both the DPLD and the GPLD. In fact, the DPLD can generate two additional chip selects called PSEL0 and PSEL1 that can be used to connect other peripheral devices to the combined system. Using the page register, these devices could be mapped into the $4000 to $7FFF range used for EPROM blocks 0 and 1. If a more complex decoding function is needed, the GPLD and its associated macrocells can be used.

CONCLUSION

Figure 6 shows the newly-enhanced system, which has plenty of free general purpose I/O to handle a large parallel interface keyboard. A number of different LCD solutions can be supported — the choices range from simple I/O driven devices to complete intelligent controller-based displays with synchronous serial or memory mapped interfaces. The system is capable of meeting next generation product specifications with room to spare for future expansion.

Highly integrated M68HC11 derivatives, such as the MC68HC711K4, can often serve as complete solutions for single-chip embedded control systems. Cost-effective designs with these devices make extensive use of on-board peripherals like the SCI, SPI, timer, and A/D converter. However, an application can outgrow the original design, and when this happens, it may be difficult to find an enhanced derivative to meet new peripheral and memory requirements.

To solve this problem, users of high performance M68HC11 can pair a ROM-less M68HC11 derivative with a WSI PSD4XX or PSD5XX programmable system device. WSI's highly integrated microcontroller peripherals can deliver a cost-effective combination of EPROM, RAM, programmable logic, digital I/O, timer, and interrupt control modules. The M68HC11/PSD combination retains many advantages of the original single-chip MCU solution while providing a flexible resource complement for future application growth.

REFERENCES

WSI *PSD Programmable Peripherals Design and Applications Handbook*.

Motorola *MC68HC11 K Technical Data Book*, Publication MC68HC11K4/D

Motorola *MC68HC11 K Programmers Reference Guide*, Publication MC68HC11KRG/D

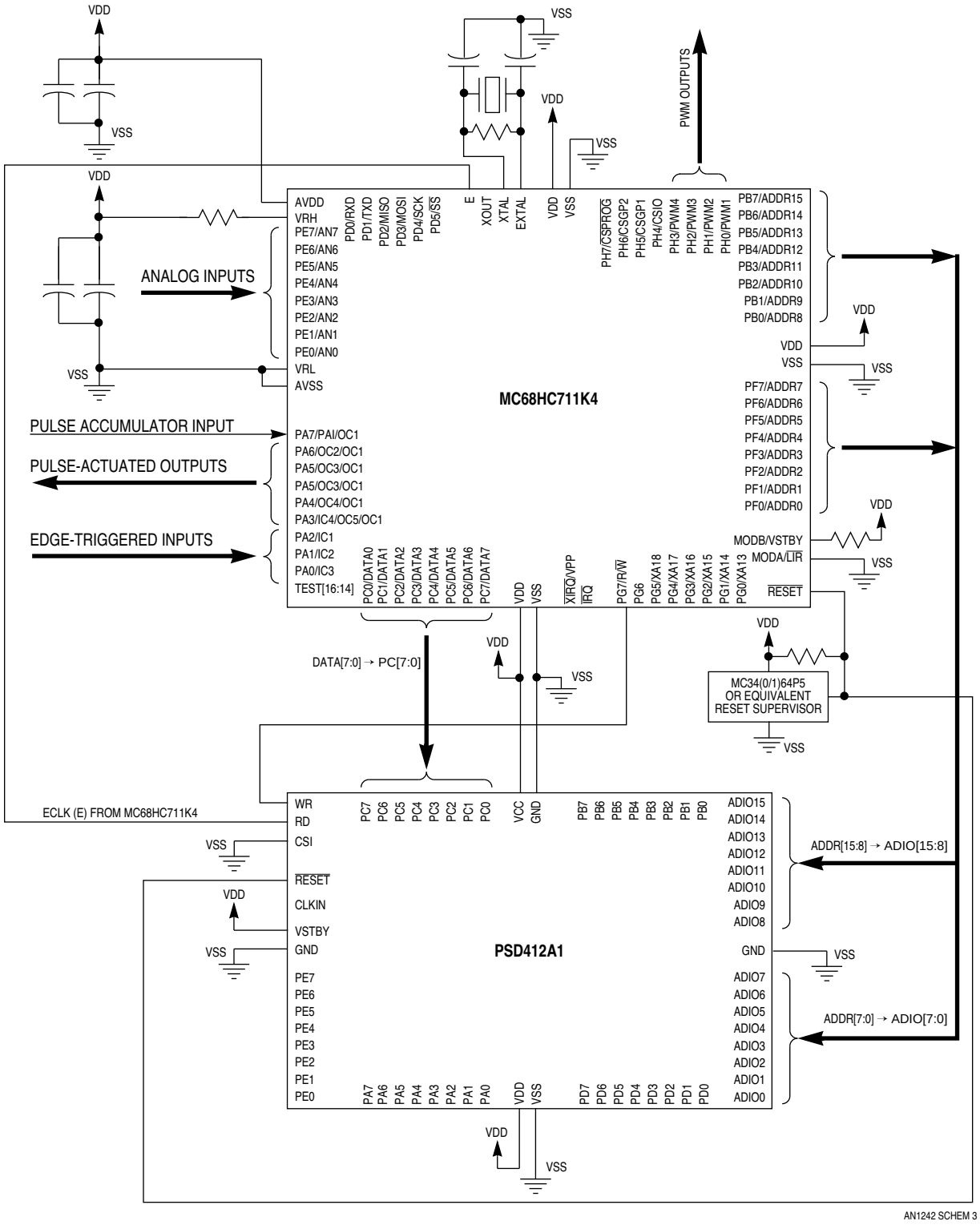

Figure 6 Keyboard and LCD Ready MC68HC11K1 + PSD412A1 System

DEVICE REFERENCE TABLES

Table 4 M68HC11 Derivatives with Non-Multiplexed Address/Data Bus

Motorola Part Number	ROM or *EPROM*	RAM (bytes)	EEPROM (bytes)	total I/O	On-Chip Peripherals	Technical Data
MC68HC11F1	0	1024	512	30	Standard[1] + 4 chip selects	MC68HC11F1/D
MC68HC11G5	16K	512	0	66	Standard +10-bit ADC + event counter	MC68HC11G5/D
MC68HC711G5	*16K*	512	0	66	Standard +10-bit ADC + event counter	MC68HC11G5/D
MC68HC11G7	24K	512	0	66	Standard +10-bit ADC + event counter	MC68HC11G5/D
MC68HC11K0	0	768	0	37	Enhanced[2] + 4 chip selects + memory expansion	MC68HC11K4/D
MC68HC11K1	0	768	640	37	Enhanced + 4 chip selects + memory expansion	MC68HC11K4/D
MC68HC11K4	24K	768	640	62	Enhanced + 4 chip selects + memory expansion	MC68HC11K4/D
MC68HC711K4	*24K*	768	640	62	Enhanced + 4 chip selects + memory expansion	MC68HC11K4/D
MC68HC11KA4	24K	768	640	51	Enhanced	MC68HC11KA4TS/D
MC68HC711KA4	*24K*	768	640	51	Enhanced	MC68HC11KA4TS/D
MC68HC11P2	32K	1024	640	62	Enhanced + 2 SCI+	MC68HC11P2/D
MC68HC711P2	*32K*	1024	640	62	Enhanced + 2 SCI+	MC68HC11P2/D

NOTES:

1. The standard peripheral complement consists of an 8-bit, 8 channel A/D converter (ADC), serial communications interface (SCI), serial peripheral interface (SPI), 16-bit timer with 3 or 4 input captures (ICs), 4 or 5 output compares (OCs), pulse accumulator, real-time interrupt, and computer operating properly (COP) watchdog monitor.
2. The enhanced peripheral complement improves on the standard peripheral complement with an SCI+ (enhanced SCI with parity generation and more flexible baud rate generator) and an enhanced SPI (additional baud rates and selectable bit shifting order)

Table 5 PSD4XX and PSD5XX Derivatives

WSI Part Number	Bus Width (Bits)	Inputs	Product Terms	Registered Macrocells	EPROM Density
PSD401A1	x 8 or x 16	37	113	8	32K x 8 or 16K x 16
PSD411A1	x 8	37	113	8	32K x 8
PSD402A1	x 8 or x 16	37	113	8	64K x 8 or 32K x 16
PSD412A1	x 8	37	113	8	64K x 8
PSD403A1	x 8 or x 16	37	113	8	128K x 8 or 64K x 16
PSD413A1	x 8	37	113	8	128K x 8
PSD401A2	x 8 or x 16	59	126	24	32K x 8 or 64K x 16
PSD411A2	x 8	59	126	24	32K x 8
PSD402A2	x 8 or x 16	59	126	24	64K x 8 or 32K x 16
PSD412A2	x 8	59	126	24	64K x 8
PSD403A2	x 8 or x 16	59	126	24	128K x 8 or 64K x 16
PSD413A2	x 8	59	126	24	128K x 8
PSD501B1	x 8 or x 16	61	140	30	32K x 8 or 16K x 16
PSD511B1	x 8	61	140	30	32K x 8
PSD502B1	x 8 or x 16	61	140	30	64K x 8 or 32K x 16
PSD512B1	x 8	61	140	30	64K x 8
PSD503B1	x 8 or x 16	61	140	30	128K x 8 or 64K x 16
PSD513B1	x 8	61	140	30	128K x 8

PSD4XX and PSD4XX devices have SRAM that can be configured as 2K x 8 or 1K x 16, 40 I/O pins, and a power management unit (PMU). PSD5XX devices have a peripheral unit consisting of four 16-bit counters/ timers, a watchdog timer, an eight-level interrupt controller, and programmable logic for memory mapping. All PSDs are available with access speeds of 90, 120, 150, or 200 nanoseconds.

TO OBTAIN ADDITIONAL PRODUCT INFORMATION:

USA/EUROPE: Motorola Literature Distribution;
P.O. Box 20912; Phoenix, Arizona 85036. 1-800-441-2447
JAPAN: Nippon Motorola Ltd.; Tatsumi-SPD-JLDC, Toshikatsu Otsuki,
6F Seibu-Butsuryu-Center, 3-14-2 Tatsumi Koto-Ku, Tokyo 135, Japan. 03-3521-8315
HONG KONG: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park,
51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852-26629298
MFAX: RMFAX0@email.sps.mot.com - TOUCHTONE (602) 244-6609
INTERNET: http://www.mot.com

MOTOROLA