Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

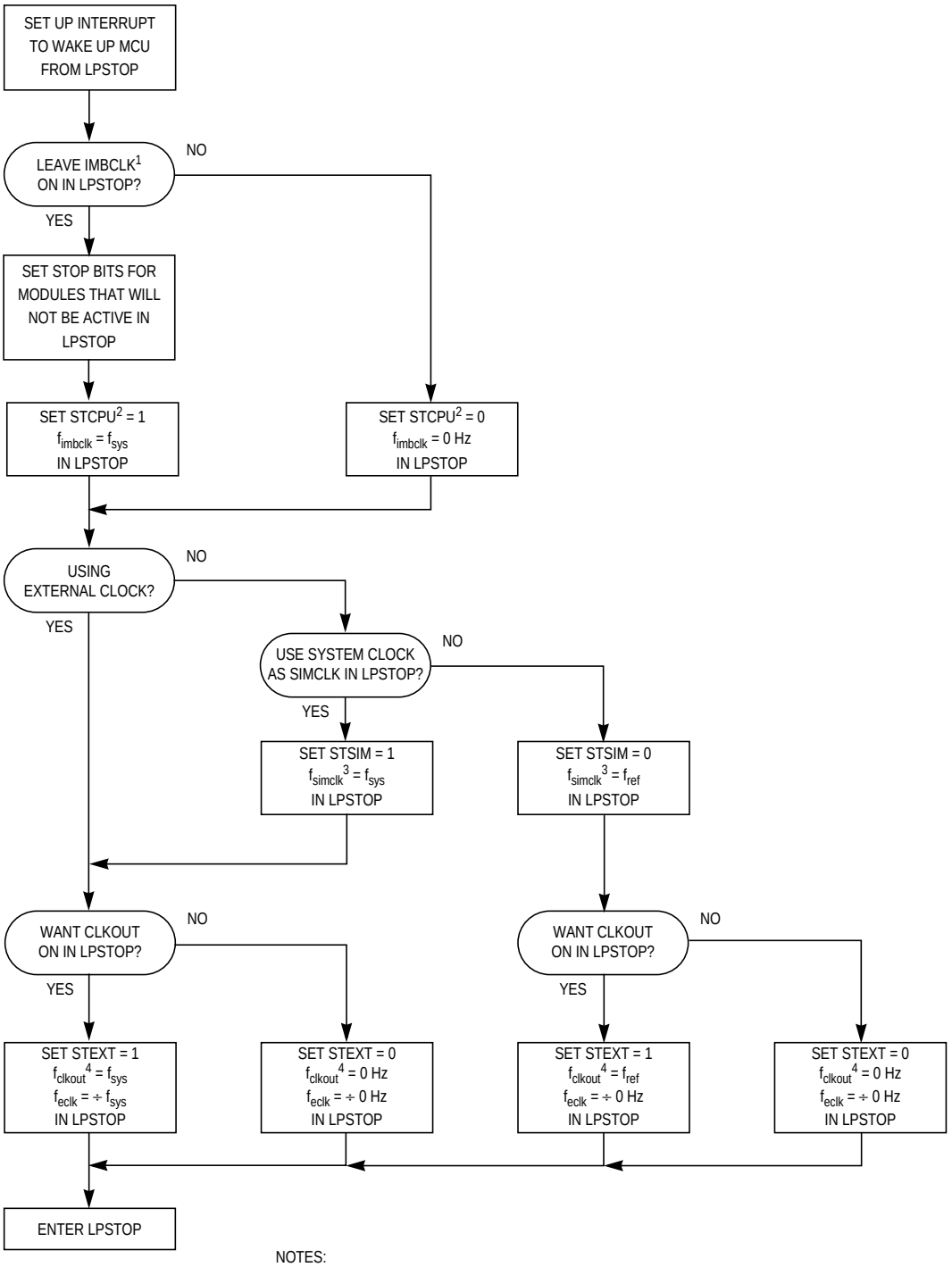
Details	
Product Status	Obsolete
Core Processor	CPU16
Core Size	16-Bit
Speed	25MHz
Connectivity	EBI/EMI, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	16
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc16z1cpv25

LIST OF ILLUSTRATIONS

LIST OF TABLES

2.3 Register Mnemonics

Mnemonic	Register
ADCMCR	ADC Module Configuration Register
ADCTEST	ADC Test Register
ADCTL[0:1]	ADC Control Registers [0:1]
ADCSTAT	ADC Status Register
CFORC	GPT Compare Force Register
CREG	SIM Test Module Control Register
CR[0:F]	QSM Command RAM [0:F]
CSBARBT	SIM Chip-Select Base Address Register Boot
CSBAR[0:10]	SIM Chip-Select Base Address Registers [0:10]
CSORBT	SIM Chip-Select Option Register Boot
CSOR[0:10]	SIM Chip-Select Option Registers [0:10]
CSPAR[0:1]	SIM Chip-Select Pin Assignment Registers [0:1]
DDRE	SIM Port E Data Direction Register
DDRF	SIM Port F Data Direction Register
DDRGP	GPT Port GP Data Direction Register
DDRM	MCCI Data Direction Register
DDRQS	QSM Port QS Data Direction Register
DREG	SIM Test Module Distributed Register
GPTMCR	GPT Module Configuration Register
GPTMTR	GPT Module Test Register
ICR	GPT Interrupt Configuration Register
ILSCI	MCCI SCI Interrupt Register
ILSPI	MCCI SPI Interrupt Register
LJSRR[0:7]	ADC Left-Justified Signed Result Registers [0:7]
LJURR[0:7]	ADC Left-Justified Unsigned Result Registers [0:7]
MIVR	MCCI Interrupt Vector Register
MMCR	MCCI Module Configuration Register
MPAR	MCCI Pin Assignment Register
MRMCR	Masked ROM Module Configuration Register
MTEST	MCCI Test Register
OC1D	GPT Output Compare 1 Action Data Register
OC1M	GPT Output Compare 1 Action Mask Register
PACNT	GPT Pulse Accumulator Counter Register
PACTL	GPT Pulse Accumulator Control Register
PEPAR	SIM Port E Pin Assignment Register
PFPAR	SIM Port F Pin Assignment Register



NOTES:

1. IMBCLK IS THE CLOCK USED BY THE CPU16L, SIML, ADC, MCCI, AND THE GPT.
2. WHEN STCPU = 1, THE CPU16L IS SHUT DOWN IN LPSTOP. ALL OTHER MODULES WILL REMAIN ACTIVE UNLESS THE STOP BITS IN THEIR MODULE CONFIGURATION REGISTERS ARE SET PRIOR TO ENTERING LPSTOP.
3. THE SIMCLK IS USED BY THE PIT, IRQ, AND INPUT BLOCKS OF THE SIML.
4. CLKOUT CONTROL DURING LPSTOP IS OVERRIDDEN BY THE EXOFF BIT IN SIMCR. IF EXOFF = 1, THE CLKOUT PIN IS ALWAYS IN A HIGH-IMPEDANCE STATE AND STEXT HAS NO EFFECT IN LPSTOP. IF EXOFF = 0, CLKOUT IS CONTROLLED BY STEXT IN LPSTOP. WHEN STCPU = 1, THE CPU16L IS DISABLED IN LPSTOP, BUT ALL OTHER MODULES REMAIN ACTIVE OR STOPPED ACCORDING TO THE SETTING.

SIML LPSTOP FLOWCHART

Figure 5-7 SIML LPSTOP Flowchart

The MRM array can be mapped to any 8-Kbyte boundary in the memory map, but must not overlap other module control registers (overlap makes the registers inaccessible). If the array overlaps the MRM register block, addresses in the register block are accessed instead of the corresponding ROM array addresses.

ROMBAH and ROMBAL can only be written while the ROM is in low-power stop mode (MRMCR STOP = 1) and the base address lock (MRMCR LOCK = 0) is disabled. LOCK can be written once only to a value of one; subsequent writes are ignored. This prevents accidental remapping of the array.

7.3 MRM Array Address Space Type

ASPC[1:0] in MRMCR determines ROM array address space type. The module can respond to both program and data space accesses or to program space accesses only. The default value of ASPC[1:0] is established during mask programming, but the value can be changed after reset if the LOCK bit in the MRMCR has not been masked to a value of one. Because the CPU16 operates in supervisor mode only, ASPC1 has no effect.

Table 7-1 shows ASPC[1:0] field encodings.

Table 7-1 ROM Array Space Field

ASPC[1:0]	State Specified
X0	Program and data accesses
X1	Program access only

Refer to **5.5.1.7 Function Codes** for more information concerning address space types and program/data space access. Refer to **4.6 Addressing Modes** for more information on addressing modes.

7.4 Normal Access

The array can be accessed by byte, word, or long word. A byte or aligned word access takes a minimum of one bus cycle (two system clocks). A long word or misaligned word access requires a minimum of two bus cycles.

Access time can be optimized for a particular application by inserting wait states into each access. The number of wait states inserted is determined by the value of WAIT[1:0] in the MRMCR. Two, three, four, or five clock accesses can be specified. The default value WAIT[1:0] is established during mask programming, but field value can be changed after reset if the LOCK bit in the MRMCR has not been masked to a value of one.

Table 7-2 shows WAIT[1:0] field encodings.

9.2.2 QSM Pin Control Registers

The QSM uses nine pins. Eight of the pins can be used for serial communication or for parallel I/O. Clearing a bit in the port QS pin assignment register (PQSPAR) assigns the corresponding pin to general-purpose I/O; setting a bit assigns the pin to the QSPI. PQSPAR does not affect operation of the SCI.

The port QS data direction register (DDRQS) determines whether pins are inputs or outputs. Clearing a bit makes the corresponding pin an input; setting a bit makes the pin an output. DDRQS affects both QSPI function and I/O function. DDQS7 determines the direction of the TXD pin only when the SCI transmitter is disabled. When the SCI transmitter is enabled, the TXD pin is an output.

The port QS data register (PORTQS) latches I/O data. PORTQS writes drive pins defined as outputs. PORTQS reads return data present on the pins. To avoid driving undefined data, first write PORTQS, then configure DDRQS.

PQSPAR and DDRQS are 8-bit registers located at the same word address. Refer to **Table 9-1** for a summary of QSM pin functions.

Table 9-1 Effect of DDRQS on QSM Pin Function

QSM Pin	QSPI Mode	DDRQS Bit	Bit State	Pin Function
MISO	Master	DDQS0	0	Serial data input to QSPI
			1	Disables data input
	Slave		0	Disables data output
			1	Serial data output from QSPI
MOSI	Master	DDQS1	0	Disables data output
			1	Serial data output from QSPI
	Slave		0	Serial data input to QSPI
			1	Disables data input
SCK[1]	Master	DDQS2	—	Clock output from QSPI
	Slave		—	Clock input to QSPI
PCS0/$\overline{SS}$	Master	DDQS3	0	Assertion causes mode fault
			1	Chip-select output
	Slave		0	QSPI slave select input
			1	Disables slave select input
PCS[1:3]	Master	DDQS[4:6]	0	Disables chip-select output
			1	Chip-select output
	Slave		0	Inactive
			1	Inactive
TXD[2]	—	DDQS7	X	Serial data output from SCI
RXD	—	None	NA	Serial data input to SCI

NOTES:
1. PQS2 is a digital I/O pin unless the SPI is enabled (SPE set in SPCR1), in which case it becomes the QSPI serial clock SCK.
2. PQS7 is a digital I/O pin unless the SCI transmitter is enabled (TE set in SCCR1), in which case it becomes the SCI serial data output TXD.

9.4.1 SCI Registers

The SCI programming model includes the QSM global and pin control registers, and four SCI registers. There are two SCI control registers (SCCR0 and SCCR1), one status register (SCSR), and one data register (SCDR). Refer to **D.6 Queued Serial Module** for register bit and field definitions.

9.4.1.1 Control Registers

SCCR0 contains the baud rate selection field. Baud rate must be set before the SCI is enabled. This register can be read or written.

SCCR1 contains a number of SCI configuration parameters, including transmitter and receiver enable bits, interrupt enable bits, and operating mode enable bits. This register can be read or written at any time. The SCI can modify the RWU bit under certain circumstances.

Changing the value of SCI control bits during a transfer may disrupt operation. Before changing register values, allow the SCI to complete the current transfer, then disable the receiver and transmitter.

9.4.1.2 Status Register

SCSR contains flags that show SCI operating conditions. These flags are cleared either by SCI hardware or by reading SCSR, then reading or writing SCDR. A long-word read can consecutively access both SCSR and SCDR. This action clears receiver status flag bits that were set at the time of the read, but does not clear TDRE or TC flags.

If an internal SCI signal for setting a status bit comes after reading the asserted status bits, but before reading or writing SCDR, the newly set status bit is not cleared. SCSR must be read again with the bit set, and SCDR must be read or written before the status bit is cleared.

Reading either byte of SCSR causes all 16 bits to be accessed, and any status bit already set in either byte is cleared on a subsequent read or write of SCDR.

9.4.1.3 Data Register

SCDR contains two data registers at the same address. The receive data register (RDR) is a read-only register that contains data received by the SCI. Data enters the receive serial shifter and is transferred to RDR. The transmit data register (TDR) is a write-only register that contains data to be transmitted. Data is first written to TDR, then transferred to the transmit serial shifter, where additional format bits are added before transmission. R[7:0]/T[7:0] contain either the first eight data bits received when SCDR is read, or the first eight data bits to be transmitted when SCDR is written. R8/T8 are used when the SCI is configured for 9-bit operation. When the SCI is configured for 8-bit operation, they have no meaning or effect.

9.4.2 SCI Pins

Two unidirectional pins, TXD (transmit data) and RXD (receive data), are associated with the SCI. TXD can be used by the SCI or for general-purpose I/O. TXD function is controlled by PQSPA7 in the port QS pin assignment register (PQSPAR) and TE in SCI control register 1 (SCCR1). The receive data (RXD) pin is dedicated to the SCI.

9.4.3 SCI Operation

The SCI can operate in polled or interrupt-driven mode. Status flags in SCSR reflect SCI conditions regardless of the operating mode chosen. The TIE, TCIE, RIE, and ILIE bits in SCCR1 enable interrupts for the conditions indicated by the TDRE, TC, RDRF, and IDLE bits in SCSR, respectively.

9.4.3.1 Definition of Terms

- Bit-Time — The time required to transmit or receive one bit of data, which is equal to one cycle of the baud frequency.
- Start Bit — One bit-time of logic zero that indicates the beginning of a data frame. A start bit must begin with a one-to-zero transition and be preceded by at least three receive time samples of logic one.
- Stop Bit — One bit-time of logic one that indicates the end of a data frame.
- Frame — A complete unit of serial information. The SCI can use 10-bit or 11-bit frames.
- Data Frame — A start bit, a specified number of data or information bits, and at least one stop bit.
- Idle Frame — A frame that consists of consecutive ones. An idle frame has no start bit.
- Break Frame — A frame that consists of consecutive zeros. A break frame has no stop bits.

9.4.3.2 Serial Formats

All data frames must have a start bit and at least one stop bit. Receiving and transmitting devices must use the same data frame format. The SCI provides hardware support for both 10-bit and 11-bit frames. The M bit in SCCR1 specifies the number of bits per frame.

The most common data frame format for NRZ serial interfaces is one start bit, eight data bits (LSB first), and one stop bit; a total of ten bits. The most common 11-bit data frame contains one start bit, eight data bits, a parity or control bit, and one stop bit. Ten-bit and 11-bit frames are shown in **Table 9-4**.

10.4.5.4 Parity Checking

The PT bit in SCCR1 selects either even (PT = 0) or odd (PT = 1) parity. PT affects received and transmitted data. The PE bit in SCCR1 determines whether parity checking is enabled (PE = 1) or disabled (PE = 0). When PE is set, the MSB of data in a frame is used for the parity function. For transmitted data, a parity bit is generated for received data; the parity bit is checked. When parity checking is enabled, the PF bit in the SCI status register (SCSR) is set if a parity error is detected.

Enabling parity affects the number of data bits in a frame, which can in turn affect frame size. **Table 10-7** shows possible data and parity formats.

Table 10-7 Effect of Parity Checking on Data Size

M	PE	Result
0	0	8 data bits
0	1	7 data bits, 1 parity bit
1	0	9 data bits
1	1	8 data bits, 1 parity bit

10.4.5.5 Transmitter Operation

The transmitter consists of a serial shifter and a parallel data register (TDR) located in the SCI data register (SCDR). The serial shifter cannot be directly accessed by the CPU16. The transmitter is double-buffered, which means that data can be loaded into the TDR while other data is shifted out. The TE bit in SCCR1 enables (TE = 1) and disables (TE = 0) the transmitter.

Shifter output is connected to the TXD pin while the transmitter is operating (TE = 1, or TE = 0 and transmission in progress). Wired-OR operation should be specified when more than one transmitter is used on the same SCI bus. The WOMS bit in SCCR1 determines whether TXD is an open-drain (wired-OR) output or a normal CMOS output. An external pull-up resistor on TXD is necessary for wired-OR operation. WOMS controls TXD function whether the pin is used by the SCI or as a general-purpose I/O pin.

Data to be transmitted is written to SCDR, then transferred to the serial shifter. The transmit data register empty (TDRE) flag in SCSR shows the status of TDR. When TDRE = 0, the TDR contains data that has not been transferred to the shifter. Writing to SCDR again overwrites the data. TDRE is set when the data in the TDR is transferred to the shifter. Before new data can be written to the SCDR, however, the processor must clear TDRE by writing to SCSR. If new data is written to the SCDR without first clearing TDRE, the data will not be transmitted.

The transmission complete (TC) flag in SCSR shows transmitter shifter state. When TC = 0, the shifter is busy. TC is set when all shifting operations are completed. TC is not automatically cleared. The processor must clear it by first reading SCSR while TC is set, then writing new data to SCDR.

Table A-16 16.78-MHz AC Timing

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 10 %, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Characteristic	Symbol	Min	Max	Unit
F1	Frequency of Operation	f	—	16.78	MHz
1	Clock Period	t_{cyc}	59.6	—	ns
1A	ECLK Period	t_{Ecyc}	476	—	ns
1B	External Clock Input Period[2]	t_{Xcyc}	59.6	—	ns
2, 3	Clock Pulse Width[3]	t_{CW}	24	—	ns
2A, 3A	ECLK Pulse Width	t_{ECW}	236	—	ns
2B, 3B	External Clock Input High/Low Time[2]	t_{XCHL}	29.8	—	ns
4, 5	CLKOUT Rise and Fall Time	t_{Crf}	—	5	ns
4A, 5A	Rise and Fall Time (All Outputs except CLKOUT)	t_{rf}	—	8	ns
4B, 5B	External Clock Input Rise and Fall Time[3]	t_{XCrf}	—	5	ns
6	Clock High to ADDR, FC, SIZE Valid[4]	t_{CHAV}	0	29	ns
7	Clock High to ADDR, Data, FC, SIZE, High Impedance	t_{CHAZx}	0	59	ns
8	Clock High to ADDR, FC, SIZE, Invalid	t_{CHAZn}	0	—	ns
9	Clock Low to $\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Asserted[4]	t_{CLSA}	2	24	ns
9A	$\overline{AS}$ to $\overline{DS}$ or $\overline{CS}$ Asserted (Read)[5]	t_{STSA}	-15	15	ns
11	ADDR, FC, SIZE Valid to $\overline{AS}$, $\overline{CS}$, (and $\overline{DS}$ Read) Asserted	t_{AVSA}	15	—	ns
12	Clock Low to $\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated	t_{CLSN}	2	29	ns
13	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated to ADDR, FC SIZE Invalid (Address Hold)	t_{SNAI}	15	—	ns
14	$\overline{AS}$, $\overline{CS}$ (and $\overline{DS}$ Read) Width Asserted	t_{SWA}	100	—	ns
14A	$\overline{DS}$, $\overline{CS}$ Width Asserted (Write)	t_{SWAW}	45	—	ns
14B	$\overline{AS}$, $\overline{CS}$ (and $\overline{DS}$ Read) Width Asserted (Fast Cycle)	t_{SWDW}	40	—	ns
15	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Width Negated[6]	t_{SN}	40	—	ns
16	Clock High to $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$ High Impedance	t_{CHSZ}	—	59	ns
17	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated to R/$\overline{W}$ High	t_{SNRN}	15	—	ns
18	Clock High to R/$\overline{W}$ High	t_{CHRH}	0	29	ns
20	Clock High to R/$\overline{W}$ Low	t_{CHRL}	0	29	ns
21	R/$\overline{W}$ High to $\overline{AS}$, $\overline{CS}$ Asserted	t_{RAAA}	15	—	ns
22	R/$\overline{W}$ Low to $\overline{DS}$, $\overline{CS}$ Asserted (Write)	t_{RASA}	70	—	ns
23	Clock High to Data Out Valid	t_{CHDO}	—	29	ns
24	Data Out Valid to Negating Edge of $\overline{AS}$, $\overline{CS}$ (Fast Write Cycle)	t_{DVASN}	15	—	ns
25	$\overline{DS}$, $\overline{CS}$ Negated to Data Out Invalid (Data Out Hold)	t_{SNDOI}	15	—	ns
26	Data Out Valid to $\overline{DS}$, $\overline{CS}$ Asserted (Write)	t_{DVSA}	15	—	ns
27	Data In Valid to Clock Low (Data Setup)[4]	t_{DICL}	5	—	ns
27A	Late $\overline{BERR}$, $\overline{HALT}$ Asserted to Clock Low (Setup Time)	t_{BELCL}	20	—	ns
28	$\overline{AS}$, $\overline{DS}$ Negated to $\overline{DSACK}$[1:0], $\overline{BERR}$, $\overline{HALT}$, $\overline{AVEC}$ Negated	t_{SNDN}	0	80	ns
29	$\overline{DS}$, $\overline{CS}$ Negated to Data In Invalid (Data In Hold)[7]	t_{SNDI}	0	—	ns
29A	$\overline{DS}$, $\overline{CS}$ Negated to Data In High Impedance[7, 8]	t_{SHDI}	—	55	ns

Table A-18 25.17-MHz AC Timing

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Characteristic	Symbol	Min	Max	Unit
F1	Frequency of Operation	f	—	25.166	MHz
1	Clock Period	t_{cyc}	39.7	—	ns
1A	ECLK Period	t_{Ecyc}	318	—	ns
1B	External Clock Input Period[2]	t_{Xcyc}	39.7	—	ns
2, 3	Clock Pulse Width[3]	t_{CW}	15	—	ns
2A, 3A	ECLK Pulse Width	t_{ECW}	155	—	ns
2B, 3B	External Clock Input High/Low Time[2]	t_{XCHL}	19.8	—	ns
4, 5	CLKOUT Rise and Fall Time	t_{Crf}	—	5	ns
4A, 5A	Rise and Fall Time (All Outputs except CLKOUT)	t_{rf}	—	8	ns
4B, 5B	External Clock Input Rise and Fall Time[3]	t_{XCrf}	—	4	ns
6	Clock High to ADDR, FC, SIZ Valid[4]	t_{CHAV}	0	19	ns
7	Clock High to ADDR, Data, FC, SIZ, High Impedance	t_{CHAZx}	0	39	ns
8	Clock High to ADDR, FC, SIZ, Invalid	t_{CHAZn}	0	—	ns
9	Clock Low to $\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Asserted[4]	t_{CLSA}	2	19	ns
9A	$\overline{AS}$ to $\overline{DS}$ or $\overline{CS}$ Asserted (Read)[5]	t_{STSA}	–10	15	ns
11	ADDR, FC, SIZE Valid to $\overline{AS}$, $\overline{CS}$, (and $\overline{DS}$ Read) Asserted	t_{AVSA}	8	—	ns
12	Clock Low to $\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated	t_{CLSN}	2	19	ns
13	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated to ADDR, FC, SIZ Invalid (Address Hold)	t_{SNAI}	8	—	ns
14	$\overline{AS}$, $\overline{CS}$ (and $\overline{DS}$ Read) Width Asserted	t_{SWA}	65	—	ns
14A	$\overline{DS}$, $\overline{CS}$ Width Asserted (Write)	t_{SWAW}	25	—	ns
14B	$\overline{AS}$, $\overline{CS}$ (and $\overline{DS}$ Read) Width Asserted (Fast Cycle)	t_{SWDW}	22	—	ns
15	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Width Negated[6]	t_{SN}	22	—	ns
16	Clock High to $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$ High Impedance	t_{CHSZ}	—	39	ns
17	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated to R/$\overline{W}$ High	t_{SNRN}	10	—	ns
18	Clock High to R/$\overline{W}$ High	t_{CHRH}	0	19	ns
20	Clock High to R/$\overline{W}$ Low	t_{CHRL}	0	19	ns
21	R/$\overline{W}$ High to $\overline{AS}$, $\overline{CS}$ Asserted	t_{RAAA}	10	—	ns
22	R/$\overline{W}$ Low to $\overline{DS}$, $\overline{CS}$ Asserted (Write)	t_{RASA}	40	—	ns
23	Clock High to Data Out Valid	t_{CHDO}	—	19	ns
24	Data Out Valid to Negating Edge of $\overline{AS}$, $\overline{CS}$ (Fast Write Cycle)	t_{DVASN}	7	—	ns
25	$\overline{DS}$, $\overline{CS}$ Negated to Data Out Invalid (Data Out Hold)	t_{SNDOI}	5	—	ns
26	Data Out Valid to $\overline{DS}$, $\overline{CS}$ Asserted (Write)	t_{DVSA}	8	—	ns
27	Data In Valid to Clock Low (Data Setup)[4]	t_{DICL}	5	—	ns
27A	Late $\overline{BERR}$, $\overline{HALT}$ Asserted to Clock Low (Setup Time)	t_{BELCL}	10	—	ns
28	$\overline{AS}$, $\overline{DS}$ Negated to $\overline{DSACK}$[1:0], $\overline{BERR}$, $\overline{HALT}$, $\overline{AVEC}$ Negated	t_{SNDN}	0	50	ns
29	$\overline{DS}$, $\overline{CS}$ Negated to Data In Invalid (Data In Hold)[7]	t_{SNDI}	0	—	ns
29A	$\overline{DS}$, $\overline{CS}$ Negated to Data In High Impedance[7, 8]	t_{SHDI}	—	45	ns

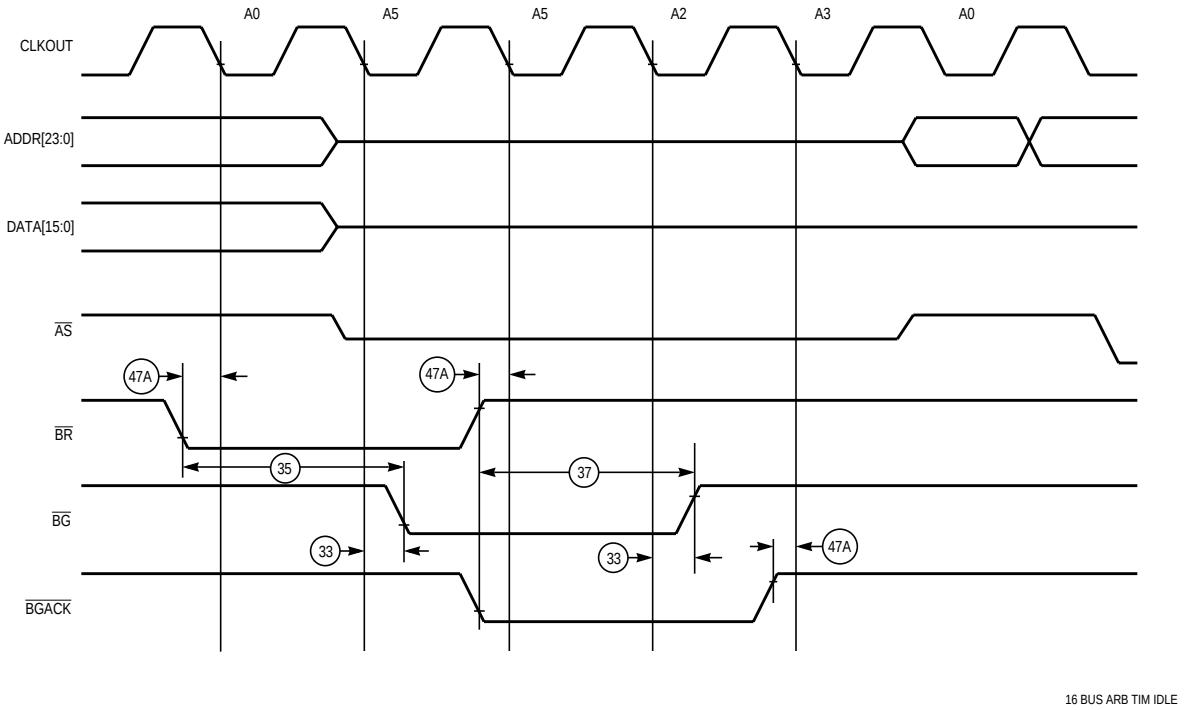


Figure A-9 Bus Arbitration Timing Diagram — Idle Bus Case

Table A-19 Low Voltage 16.78-MHz Background Debug Mode Timing

(V_{DD} and V_{DDSYN} = 2.7 to 3.6Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Characteristic	Symbol	Min	Max	Unit
B0	DSI Input Setup Time	t_{DSISU}	15	—	ns
B1	DSI Input Hold Time	t_{DSIH}	15	—	ns
B2	DSCLK Setup Time	t_{DSCSU}	15	—	ns
B3	DSCLK Hold Time	t_{DSCH}	15	—	ns
B4	DSO Delay Time	t_{DSOD}	—	35	ns
B5	DSCLK Cycle Time	t_{DSCCYC}	2	—	t_{cyc}
B6	CLKOUT High to FREEZE Asserted/Negated	t_{FRZAN}	—	50	ns
B7	CLKOUT High to IPIPE1 High Impedance	t_{IFZ}	—	50	ns
B8	CLKOUT High to IPIPE1 Valid	t_{IF}	—	50	ns
B9	DSCLK Low Time	t_{DSCLO}	1	—	t_{cyc}
B10	IPIPE1 High Impedance to FREEZE Asserted	t_{IPFA}	TBD	—	t_{cyc}
B11	FREEZE Negated to IPIPE[0:1] Active	t_{FRIP}	TBD	—	t_{cyc}

NOTES:
1. All AC timing is shown with respect to V_{IH}/V_{IL} levels unless otherwise noted.

Table A-20 16.78-MHz Background Debug Mode Timing

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Characteristic	Symbol	Min	Max	Unit
B0	DSI Input Setup Time	t_{DSISU}	15	—	ns
B1	DSI Input Hold Time	t_{DSIH}	10	—	ns
B2	DSCLK Setup Time	t_{DSCSU}	15	—	ns
B3	DSCLK Hold Time	t_{DSCH}	10	—	ns
B4	DSO Delay Time	t_{DSOD}	—	25	ns
B5	DSCLK Cycle Time	t_{DSCCYC}	2	—	t_{cyc}
B6	CLKOUT High to FREEZE Asserted/Negated	t_{FRZAN}	—	50	ns
B7	CLKOUT High to IPIPE1 High Impedance	t_{IFZ}	—	TBD	ns
B8	CLKOUT High to IPIPE1 Valid	t_{IF}	—	TBD	ns
B9	DSCLK Low Time	t_{DSCLO}	1	—	t_{cyc}
B10	IPIPE1 High Impedance to FREEZE Asserted	t_{IPFA}	TBD	—	t_{cyc}
B11	FREEZE Negated to IPIPE[0:1] Active	t_{FRIP}	TBD	—	t_{cyc}

NOTES:
1. All AC timing is shown with respect to V_{IH}/V_{IL} levels unless otherwise noted.

Table A-28 QSPI Timing

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 5% for 16.78 MHz, 10% for 20/25 MHz, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)[1]

Num	Function	Symbol	Min	Max	Unit
1	Operating Frequency Master Slave	f_{op}	DC DC	1/4 1/4	f_{sys} f_{sys}
2	Cycle Time Master Slave	t_{qcyc}	4 4	510 —	t_{cyc} t_{cyc}
3	Enable Lead Time Master Slave	t_{lead}	2 2	128 —	t_{cyc} t_{cyc}
4	Enable Lag Time Master Slave	t_{lag}	— 2	1/2 —	SCK t_{cyc}
5	Clock (SCK) High or Low Time Master Slave[2]	t_{sw}	2 t_{cyc} – 60 2 t_{cyc} – n	255 t_{cyc} —	ns ns
6	Sequential Transfer Delay Master Slave (Does Not Require Deselect)	t_{td}	17 13	8192 —	t_{cyc} t_{cyc}
7	Data Setup Time (Inputs) Master Slave	t_{su}	30 20	— —	ns ns
8	Data Hold Time (Inputs) Master Slave	t_{hi}	0 20	— —	ns ns
9	Slave Access Time	t_a	—	1	t_{cyc}
10	Slave MISO Disable Time	t_{dis}	—	2	t_{cyc}
11	Data Valid (after SCK Edge) Master Slave	t_v	— —	50 50	ns ns
12	Data Hold Time (Outputs) Master Slave	t_{ho}	0 0	— —	ns ns
13	Rise Time Input Output	 t_{ri} t_{ro}	 — —	 2 30	 µs ns
14	Fall Time Input Output	 t_{fi} t_{fo}	 — —	 2 30	 µs ns

NOTES:
1. All AC timing is shown with respect to V_{IH}/V_{IL} levels unless otherwise noted.
2. For high time, n = External SCK rise time; for low time, n = External SCK fall time.

D.2 System Integration Module

Table D-2 shows the SIM address map.

Table D-2 SIM Address Map

Address[1]	15 8	7 0
$YFFA00	SIM Module Configuration Register (SIMCR)	
$YFFA02	SIM Test Register (SIMTR)	
$YFFA04	Clock Synthesizer Control Register (SYNCR)	
$YFFA06	Not Used	Reset Status Register (RSR)
$YFFA08	SIM Test Register E (SIMTRE)	
$YFFA0A	Not Used	Not Used
$YFFA0C	Not Used	Not Used
$YFFA0E	Not Used	Not Used
$YFFA10	Not Used	Port E Data Register 0 (PORTE0)
$YFFA12	Not Used	Port E Data Register 1(PORTE1)
$YFFA14	Not Used	Port E Data Direction Register (DDRE)
$YFFA16	Not Used	Port E Pin Assignment Register (PEPAR)
$YFFA18	Not Used	Port F Data Register 0 (PORTF0)
$YFFA1A	Not Used	Port F Data Register 1 (PORTF1)
$YFFA1C	Not Used	Port F Data Direction Register (DDRF)
$YFFA1E	Not Used	Port F Pin Assignment Register (PFPAR)
$YFFA20	Not Used	System Protection Control Register (SYPCR)
$YFFA22	Periodic Interrupt Control Register (PICR)	
$YFFA24	Periodic Interrupt Timer Register (PITR)	
$YFFA26	Not Used	Software Watchdog Service Register (SWSR)
$YFFA28	Not Used	
$YFFA2A	Not Used	
$YFFA2C	Not Used	
$YFFA2E	Not Used	
$YFFA30	Test Module Master Shift A Register (TSTMSRA)	
$YFFA32	Test Module Master Shift B Register (TSTMSRB)	
$YFFA34	Test Module Shift Count Register (TSTSC)	
$YFFA36	Test Module Repetition Counter Register (TSTRC)	
$YFFA38	Test Module Control Register (CREG)	
$YFFA3A	Test Module Distributed Register (DREG)	
$YFFA3C	Not Used	
$YFFA3E	Not Used	
$YFFA40	Not Used	Port C Data Register (PORTC)
$YFFA42	Not Used	Not Used
$YFFA44	Chip-Select Pin Assignment Register 0 (CSPAR0)	
$YFFA46	Chip-Select Pin Assignment Register 1 (CSPAR1)	
$YFFA48	Chip-Select Base Address Register Boot (CSBARBT)	
$YFFA4A	Chip-Select Option Register Boot (CSORBT)	
	Chip-Select Base Address Register 0 (CSBAR0)	

D.6 Queued Serial Module

Table D-31 QSM Address Map

Address[1]	15 8	7 0
$YFFC00	QSM Module Configuration Register (QSMCR)	
$YFFC02	QSM Test Register (QTEST)	
$YFFC04	QSM Interrupt Level Register (QILR)	QSM Interrupt Vector Register (QIVR)
$YFFC06	Not Used	
$YFFC08	SCI Control 0 Register (SCCR0)	
$YFFC0A	SCI Control 1 Register (SCCR1)	
$YFFC0C	SCI Status Register (SCSR)	
$YFFC0E	SCI Data Register (SCDR)	
$YFFC10	Not Used	
$YFFC12	Not Used	
$YFFC14	Not Used	Port QS Data Register (PORTQS)
$YFFC16	Port QS Pin Assignment Register (PQSPAR)	Port QS Data Direction Register (DDRQS)
$YFFC18	SPI Control Register 0 (SPCR0)	
$YFFC1A	SPI Control Register 1 (SPCR1)	
$YFFC1C	SPI Control Register 2 (SPCR2)	
$YFFC1E	SPI Control Register 3 (SPCR3)	SPI Status Register (SPSR)
$YFFC20 – $YFFCFF	Not Used	
$YFFD00 – $YFFD1F	Receive RAM (RR[0:F])	
$YFFD20 – $YFFD3F	Transmit RAM (TR[0:F])	
$YFFD40 – $YFFD4F	Command RAM (CR[0:F])	

NOTES:
1. Y = M111, where M is the logic state of the module mapping (MM) bit in the SIMCR.

D.6.1 QSM Configuration Register

QSMCR — QSM Configuration Register **$YFFC00**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STOP	FRZ1	FRZ0	NOT USED					SUPV	NOT USED			IARB[3:0]			
RESET:															
0	0	0						1				0	0	0	0

QSMCR bits enable stop and freeze modes, and determine the arbitration priority of QSM interrupt requests.

ILQSPI[2:0] — Interrupt Level for QSPI

When an interrupt request is made, the ILQSPI value determines the priority level of all QSPI interrupts. When a request is acknowledged, the QSM compares this value to a mask value supplied by the CPU16 to determine whether to respond. ILQSPI must have a value in the range $0 (interrupts disabled) to $7 (highest priority).

ILSCI[2:0] — Interrupt Level for SCI

When an interrupt request is made, the ILSCI value determines the priority level of all SCI interrupts. When a request is acknowledged, the QSM compares this value to a mask value supplied by the CPU16 to determine whether to respond. The field must have a value in the range $0 (interrupts disabled) to $7 (highest priority).

If ILQSPI[2:0] and ILSCI[2:0] have the same non-zero value, and both submodules simultaneously request interrupt service, the QSPI takes priority over the SCI.

INTV[7:0] — Interrupt Vector Number

The value of INTV[7:1] is used for both QSPI and SCI interrupt requests; the value of INTV0 used during an interrupt acknowledge cycle is supplied by the QSM. INTV0 is at logic level zero during an SCI interrupt and at logic level one during a QSPI interrupt. A write to INTV0 has no effect. Reads of INTV0 return a value of one. At reset, QIVR is initialized to $0F, the uninitialized interrupt vector number. To use interrupt-driven serial communication, a user-defined vector number must be written to QIVR.

D.6.4 SCI Control Register

SCCR0 — SCI Control Register 0 **$YFFC08**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED			SCBR[12:0]												
RESET:			0	0	0	0	0	0	0	0	0	0	1	0	0

SCCR0 contains the SCI baud rate selection field. Baud rate must be set before the SCI is enabled. The CPU16 can read and write SCCR0 at any time. Changing the value of SCCR0 bits during a transfer operation disrupts operation.

Bits [15:13] — Not Implemented

SCBR[12:0] — SCI Baud Rate

SCI baud rate is programmed by writing a 13-bit value to this field. Writing a value of zero to SCBR disables the baud rate generator. The baud clock rate is calculated as follows:

$$\text{SCI Baud Rate} = \frac{f_{sys}}{32 \times \text{SCBR[12:0]}}$$

or

$$\text{SCBR[12:0]} = \frac{f_{sys}}{32 \times \text{SCI Baud Rate Desired}}$$

where SCBR[12:0] is in the range of 1 to 8191.

D.7.13 SPI Control Register

SPCR — SPI Control Register **$YFFC38**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPIE	SPE	WOMP	MSTR	CPOL	CPHA	LSBF	SIZE	SPBR[7:0]							
RESET:															
0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0

The SPCR contains parameters for configuring the SPI. The register can be read or written at any time.

SPIE — SPI Interrupt Enable
- 0 = SPI interrupts disabled.
- 1 = SPI interrupts enabled.

SPE — SPI Enable
- 0 = SPI is disabled.
- 1 = SPI is enabled.

WOMP — Wired-OR Mode for SPI Pins
- 0 = Outputs have normal CMOS drivers.
- 1 = Pins designated for output by MDDR have open-drain drivers, regardless of whether the pins are used as SPI outputs or for general-purpose I/O, and regardless of whether the SPI is enabled.

MSTR — Master/Slave Mode Select
- 0 = SPI is a slave device.
- 1 = SPI is system master.

CPOL — Clock Polarity
- 0 = The inactive state value of SCK is logic level zero.
- 1 = The inactive state value of SCK is logic level one.

CPOL is used to determine the inactive state of the serial clock (SCK). It is used with CPHA to produce a desired clock/data relationship between master and slave devices.

CPHA — Clock Phase
- 0 = Data captured on the leading edge of SCK and changed on the trailing edge of SCK.
- 1 = Data is changed on the leading edge of SCK and captured on the trailing edge of SCK.

CPHA determines which edge of SCK causes data to change and which edge causes data to be captured. CPHA is used with CPOL to produce a desired clock/data relationship between master and slave devices.

LSBF — Least Significant Bit First
- 0 = Serial data transfer starts with LSB.
- 1 = Serial data transfer starts with MSB.

OC1M[5:1] — OC1 Mask Field
OC1M[5:1] correspond to OC[5:1].
0 = Corresponding output compare pin is not affected by OC1 compare.
1 = Corresponding output compare pin is affected by OC1 compare.

OC1D[5:1] — OC1 Data Field
OC1D[5:1] correspond to OC[5:1].
0 = If OC1 mask bit is set, clear the corresponding output compare pin on OC1 match.
1 = If OC1 mask bit is set, the set corresponding output compare pin on OC1 match.

D.8.6 Timer Counter Register

TCNT — Timer Counter Register **$YFF90A**

TCNT is the 16-bit free-running counter associated with the input capture, output compare, and pulse accumulator functions of the GPT module.

D.8.7 Pulse Accumulator Control Register/Counter

PACTL/PACNT — Pulse Accumulator Control Register/Counter **$YFF90C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAIS	PAEN	PAMOD	PEDGE	PCLKS	I4/O5	PACLK[1:0]		PULSE ACCUMULATOR COUNTER							
RESET:															
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PACTL enables the pulse accumulator and selects either event counting or gated mode. In event counting mode, PACNT is incremented each time an event occurs. In gated mode, it is incremented by an internal clock.

PAIS — PAI Pin State (Read Only)

PAEN — Pulse Accumulator Enable
0 = Pulse accumulator disabled.
1 = Pulse accumulator enabled.

PAMOD — Pulse Accumulator Mode
0 = External event counting.
1 = Gated time accumulation.

PEDGE — Pulse Accumulator Edge Control
The effects of PAMOD and PEDGE are shown in **Table D-44**.

```
        DC.W  BDM     ;241  User Defined Interrupt Vector 186
        DC.W  BDM     ;242  User Defined Interrupt Vector 187
        DC.W  BDM     ;243  User Defined Interrupt Vector 188
        DC.W  BDM     ;244  User Defined Interrupt Vector 189
        DC.W  BDM     ;245  User Defined Interrupt Vector 190
        DC.W  BDM     ;246  User Defined Interrupt Vector 191
        DC.W  BDM     ;247  User Defined Interrupt Vector 192
        DC.W  BDM     ;248  User Defined Interrupt Vector 193
        DC.W  BDM     ;249  User Defined Interrupt Vector 194
        DC.W  BDM     ;250  User Defined Interrupt Vector 195
        DC.W  BDM     ;251  User Defined Interrupt Vector 196
        DC.W  BDM     ;252  User Defined Interrupt Vector 197
        DC.W  BDM     ;253  User Defined Interrupt Vector 198
        DC.W  BDM     ;254  User Defined Interrupt Vector 199
        DC.W  BDM     ;255  User Defined Interrupt Vector 200
```

E.1.4 INITSYS.ASM

```
*       Title : INITSYS
*       Description : Initialize & configure system including
*                     the Software Watchdog and System Clock.
*************************************************************************

INITSYS:                        ;give initial values for extension registers
                                ;and initialize system clock and COP
        LDAB    #$0F
        TBEK                    ; point EK to bank F for register access
        LDAB    #$00
        TBXK                    ; point XK to bank 0
        TBYK                    ; point YK to bank 0
        TBZK                    ; point ZK to bank 0

        LDD     #$0003          ; at reset, the CSBOOT block size is 512k.
        STD     CSBARBT         ; this line sets the block size to 64k
        LDD     #$3830; async, both byte, R/W, AS, Zero WS, S/U SP, IPL all,
                                ;AVEC off
        STD     CSORBT          ;
        LDAA    #$7F            ; w=0, x=1, y=111111
        STAA    SYNCR           ; set system clock to 16.78 Mhz

        CLR     SYPCR           ; turn COP (software watchdog) off,
                                ; since COP is on after reset
```

E.1.5 INITRAM.ASM

```
*       Title : INITRAM
*       Description : Initialize the HC16's 1K internal SRAM
*                     (put SRAM in memory map at $10000, bank 1)
*                     and set the stack inside it.
*************************************************************************

INITRAM:                        ;initialize internal SRAM and stack
        LDD     #$0001
        STD     RAMBAH          ; store high ram array, bank 1
        LDD     #$0000
```