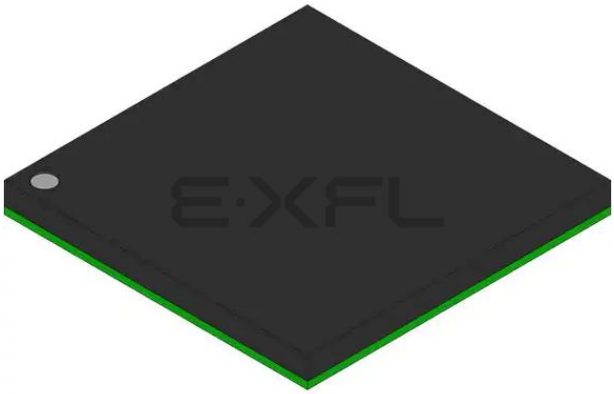


Welcome to E-XFL.COM

Understanding Embedded - DSP (Digital Signal Processors)

Embedded - DSP (Digital Signal Processors) are specialized microprocessors designed to perform complex mathematical computations on digital signals in real-time. Unlike general-purpose processors, DSPs are optimized for high-speed numeric processing tasks, making them ideal for applications that require efficient and precise manipulation of digital data. These processors are fundamental in converting and processing signals in various forms, including audio, video, and communication signals, ensuring that data is accurately interpreted and utilized in embedded systems.

Applications of Embedded - DSP (Digital Signal Processors)

Details	
Product Status	Active
Type	Fixed Point
Interface	Host Interface, SSI, SCI
Clock Rate	220MHz
Non-Volatile Memory	ROM (576B)
On-Chip RAM	576kB
Voltage - I/O	3.30V
Voltage - Core	1.60V
Operating Temperature	-40°C ~ 100°C
Mounting Type	Surface Mount
Package / Case	196-LBGA
Supplier Device Package	196-MAPBGA (15x15)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=dsp56321vl220

Product Documentation

The documents listed in **Table 2** are required for a complete description of the DSP56321 device and are necessary to design properly with the part. Documentation is available from a local Freescale distributor, a Freescale semiconductor sales office, or a Freescale Semiconductor Literature Distribution Center. For documentation updates, visit the Freescale DSP website. See the contact information on the back cover of this document.

Table 2. DSP56321 Documentation

Name	Description	Order Number
DSP56321 Reference Manual	Detailed functional description of the DSP56321 memory configuration, operation, and register programming	DSP56321RM
DSP56300 Family Manual	Detailed description of the DSP56300 family processor core and instruction set	DSP56300FM
Application Notes	Documents describing specific applications or optimized device operation including code examples	See the DSP56321 product website

1.4 External Memory Expansion Port (Port A)

Note: When the DSP56321 enters a low-power standby mode (stop or wait), it releases bus mastership and tri-states the relevant Port A signals: A[0–17], D[0–23], AA[0–3], $\overline{RD}$, $\overline{WR}$, $\overline{BB}$.

1.4.1 External Address Bus

Table 1-5. External Address Bus Signals

Signal Name	Type	State During Reset, Stop, or Wait	Signal Description
A[0–17]	Output	Tri-stated	**Address Bus**—When the DSP is the bus master, A[0–17] are active-high outputs that specify the address for external program and data memory accesses. Otherwise, the signals are tri-stated. To minimize power dissipation, A[0–17] do not change state when external memory spaces are not being accessed.

1.4.2 External Data Bus

Table 1-6. External Data Bus Signals

Signal Name	Type	State During Reset	State During Stop or Wait	Signal Description
D[0–23]	Input/ Output	Ignored Input	Last state: *Input*: Ignored *Output*: Last value	**Data Bus**—When the DSP is the bus master, D[0–23] are active-high, bidirectional input/outputs that provide the bidirectional data bus for external program and data memory accesses. Otherwise, D[0–23] drivers are tri-stated. If the last state is output, these lines have weak keepers to maintain the last output state if all drivers are tri-stated.

1.4.3 External Bus Control

Table 1-7. External Bus Control Signals

Signal Name	Type	State During Reset, Stop, or Wait	Signal Description
AA[0–3]	Output	Tri-stated	**Address Attribute**—When defined as AA, these signals can be used as chip selects or additional address lines. The default use defines a priority scheme under which only one AA signal can be asserted at a time. Setting the AA priority disable (APD) bit (Bit 14) of the Operating Mode Register, the priority mechanism is disabled and the lines can be used together as four external lines that can be decoded externally into 16 chip select signals.
$\overline{RD}$	Output	Tri-stated	**Read Enable**—When the DSP is the bus master, $\overline{RD}$ is an active-low output that is asserted to read external memory on the data bus (D[0–23]). Otherwise, $\overline{RD}$ is tri-stated.
$\overline{WR}$	Output	Tri-stated	**Write Enable**—When the DSP is the bus master, $\overline{WR}$ is an active-low output that is asserted to write external memory on the data bus (D[0–23]). Otherwise, the signals are tri-stated.

Table 1-10. Host Interface (Continued)

Signal Name	Type	State During Reset[1,2]	Signal Description
$\overline{\text{HDS}}$/HDS	Input	Ignored Input	**Host Data Strobe**—When the HI08 is programmed to interface with a single-data-strobe host bus and the HI function is selected, this signal is the host data strobe (HDS) Schmitt-trigger input. The polarity of the data strobe is programmable but is configured as active-low ($\overline{\text{HDS}}$) following reset.
$\overline{\text{HWR}}$/HWR	Input		**Host Write Data**—When the HI08 is programmed to interface with a double-data-strobe host bus and the HI function is selected, this signal is the host write data strobe (HWR) Schmitt-trigger input. The polarity of the data strobe is programmable but is configured as active-low ($\overline{\text{HWR}}$) following reset.
PB12	Input or Output		**Port B 12**—When the HI08 is configured as GPIO through the HI08 Port Control Register, this signal is individually programmed as an input or output through the HI08 Data Direction Register.
$\overline{\text{HREQ}}$/HREQ	Output	Ignored Input	**Host Request**—When the HI08 is programmed to interface with a single host request host bus and the HI function is selected, this signal is the host request (HREQ) output. The polarity of the host request is programmable but is configured as active-low ($\overline{\text{HREQ}}$) following reset. The host request may be programmed as a driven or open-drain output.
$\overline{\text{HTRQ}}$/HTRQ	Output		**Transmit Host Request**—When the HI08 is programmed to interface with a double host request host bus and the HI function is selected, this signal is the transmit host request (HTRQ) output. The polarity of the host request is programmable but is configured as active-low ($\overline{\text{HTRQ}}$) following reset. The host request may be programmed as a driven or open-drain output.
PB14	Input or Output		**Port B 14**—When the HI08 is configured as GPIO through the HI08 Port Control Register, this signal is individually programmed as an input or output through the HI08 Data Direction Register.
$\overline{\text{HACK}}$/HACK	Input	Ignored Input	**Host Acknowledge**—When the HI08 is programmed to interface with a single host request host bus and the HI function is selected, this signal is the host acknowledge (HACK) Schmitt-trigger input. The polarity of the host acknowledge is programmable but is configured as active-low ($\overline{\text{HACK}}$) after reset.
$\overline{\text{HRRQ}}$/HRRQ	Output		**Receive Host Request**—When the HI08 is programmed to interface with a double host request host bus and the HI function is selected, this signal is the receive host request (HRRQ) output. The polarity of the host request is programmable but is configured as active-low ($\overline{\text{HRRQ}}$) after reset. The host request may be programmed as a driven or open-drain output.
PB15	Input or Output		**Port B 15**—When the HI08 is configured as GPIO through the HI08 Port Control Register, this signal is individually programmed as an input or output through the HI08 Data Direction Register.

Notes: 1. In the Stop state, the signal maintains the last state as follows:
- If the last state is input, the signal is an ignored input.
- If the last state is output, these lines have weak keepers that maintain the last output state even if the drivers are tri-stated.

2. The Wait processing state does not affect the signal state.

1.7 Enhanced Synchronous Serial Interface 0 (ESSI0)

Two synchronous serial interfaces (ESSI0 and ESSI1) provide a full-duplex serial port for serial communication with a variety of serial devices, including one or more industry-standard codecs, other DSPs, microprocessors, and peripherals that implement the Freescale serial peripheral interface (SPI).

Table 1-11. Enhanced Synchronous Serial Interface 0

Signal Name	Type	State During Reset[1,2]	Signal Description
SC00	Input or Output	Ignored Input	**Serial Control 0**—For asynchronous mode, this signal is used for the receive clock I/O (Schmitt-trigger input). For synchronous mode, this signal is used either for transmitter 1 output or for serial I/O flag 0.
PC0	Input or Output		**Port C 0**—The default configuration following reset is GPIO input PC0. When configured as PC0, signal direction is controlled through the Port C Direction Register. The signal can be configured as ESSI signal SC00 through the Port C Control Register.
SC01	Input/Output	Ignored Input	**Serial Control 1**—For asynchronous mode, this signal is the receiver frame sync I/O. For synchronous mode, this signal is used either for transmitter 2 output or for serial I/O flag 1.
PC1	Input or Output		**Port C 1**—The default configuration following reset is GPIO input PC1. When configured as PC1, signal direction is controlled through the Port C Direction Register. The signal can be configured as an ESSI signal SC01 through the Port C Control Register.
SC02	Input/Output	Ignored Input	**Serial Control Signal 2**—The frame sync for both the transmitter and receiver in synchronous mode, and for the transmitter only in asynchronous mode. When configured as an output, this signal is the internally generated frame sync signal. When configured as an input, this signal receives an external frame sync signal for the transmitter (and the receiver in synchronous operation).
PC2	Input or Output		**Port C 2**—The default configuration following reset is GPIO input PC2. When configured as PC2, signal direction is controlled through the Port C Direction Register. The signal can be configured as an ESSI signal SC02 through the Port C Control Register.
SCK0	Input/Output	Ignored Input	**Serial Clock**—Provides the serial bit rate clock for the ESSI. The SCK0 is a clock input or output, used by both the transmitter and receiver in synchronous modes or by the transmitter in asynchronous modes. Although an external serial clock can be independent of and asynchronous to the DSP system clock, it must exceed the minimum clock cycle time of 6T (that is, the system clock frequency must be at least three times the external ESSI clock frequency). The ESSI needs at least three DSP phases inside each half of the serial clock.
PC3	Input or Output		**Port C 3**—The default configuration following reset is GPIO input PC3. When configured as PC3, signal direction is controlled through the Port C Direction Register. The signal can be configured as an ESSI signal SCK0 through the Port C Control Register.
SRD0	Input	Ignored Input	**Serial Receive Data**—Receives serial data and transfers the data to the ESSI Receive Shift Register. SRD0 is an input when data is received.
PC4	Input or Output		**Port C 4**—The default configuration following reset is GPIO input PC4. When configured as PC4, signal direction is controlled through the Port C Direction Register. The signal can be configured as an ESSI signal SRD0 through the Port C Control Register.

2.2 Thermal Characteristics

Table 2-2. Thermal Characteristics

Thermal Resistance Characteristic	Symbol	MAP-BGA Value	Unit
Junction-to-ambient, natural convection, single-layer board (1s)[1,2]	$R_{\theta JA}$	44	°C/W
Junction-to-ambient, natural convection, four-layer board (2s2p)[1,3]	$R_{\theta JMA}$	25	°C/W
Junction-to-ambient, @200 ft/min air flow, single-layer board (1s)[1,3]	$R_{\theta JMA}$	35	°C/W
Junction-to-ambient, @200 ft/min air flow, four-layer board (2s2p)[1,3]	$R_{\theta JMA}$	22	°C/W
Junction-to-board[4]	$R_{\theta JB}$	13	°C/W
Junction-to-case thermal resistance[5]	$R_{\theta JC}$	7	°C/W

Notes:
1. Junction temperature is a function of die size, on-chip power dissipation, package thermal resistance, mounting site (board) temperature, ambient temperature, air flow, power dissipation of other components on the board, and board thermal resistance.
2. Per SEMI G38-87 and JEDEC JESD51-2 with the single-layer board horizontal.
3. Per JEDEC JESD51-6 with the board horizontal.
4. Thermal resistance between the die and the printed circuit board per JEDEC JESD51-8. Board temperature is measured on the top surface of the board near the package.
5. Thermal resistance between the die and the case top surface as measured by the cold plate method (MIL SPEC-883 Method 1012.1).

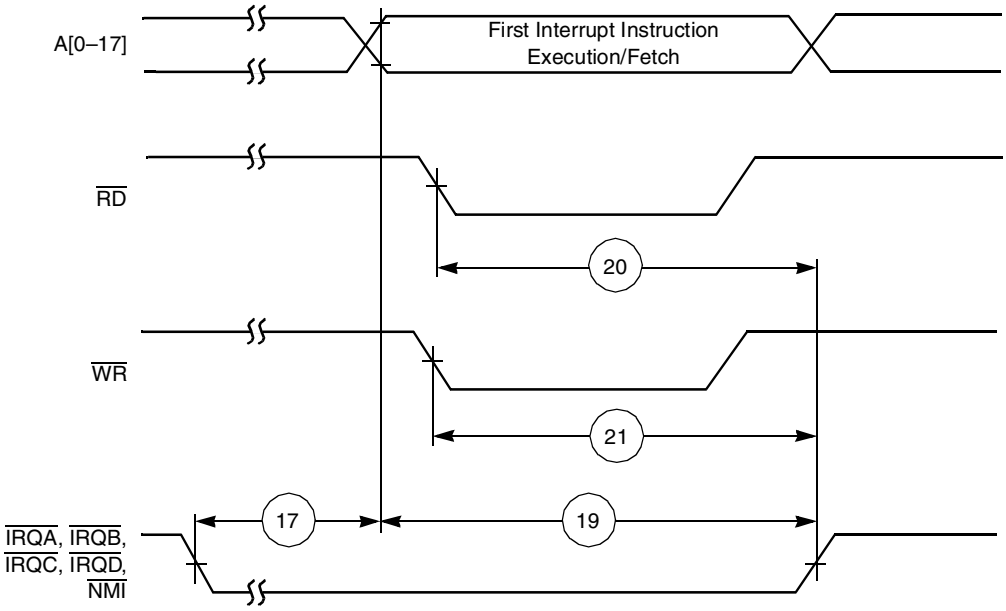
2.3 DC Electrical Characteristics

Table 2-3. DC Electrical Characteristics[7]

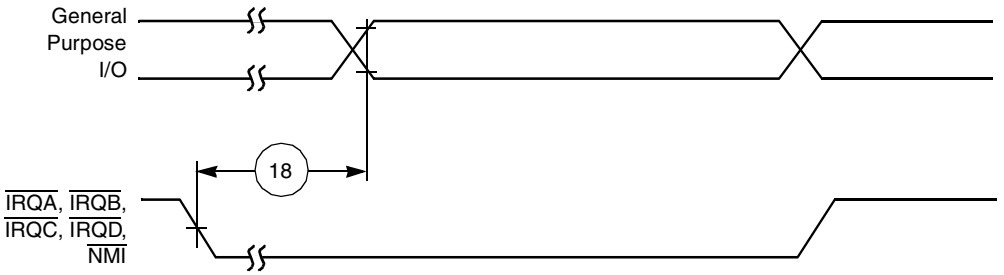
Characteristics	Symbol	Min	Typ	Max	Unit
Supply voltage[1]: • Core (V_{CCQL}) • I/O (V_{CCQH}, V_{CCA}, V_{CCD}, V_{CCC}, V_{CCH}, and V_{CCS})		 1.5 3.0	 1.6 3.3	 1.7 3.6	 V V
Input high voltage • D[0–23], $\overline{BG}$, $\overline{BB}$, $\overline{TA}$ • MOD/$\overline{IRQ}$[2] $\overline{RESET}$, PINIT/$\overline{NMI}$ and all JTAG/ESSI/SCI/Timer/HI08 pins • EXTAL[9]	 V_{IH} V_{IHP} V_{IHX}	 2.0 2.0 $0.8 \times V_{CCQH}$	 — — —	 $V_{CCQH} + 0.3$ $V_{CCQH} + 0.3$ V_{CCQH}	 V V V
Input low voltage • D[0–23], $\overline{BG}$, $\overline{BB}$, $\overline{TA}$, MOD/$\overline{IRQ}$[2], $\overline{RESET}$, PINIT • All JTAG/ESSI/SCI/Timer/HI08 pins • EXTAL[9]	 V_{IL} V_{ILP} V_{ILX}	 –0.3 –0.3 –0.3	 — — —	 0.8 0.8 $0.2 \times V_{CCQH}$	 V V V
Input leakage current	I_{IN}	–10	—	10	µA
High impedance (off-state) input current (@ 2.4 V / 0.4 V)	I_{TSI}	–10	—	10	µA
Output high voltage[8] • TTL (I_{OH} = –0.4 mA)[6] • CMOS (I_{OH} = –10 µA)[6]	V_{OH}	 2.4 $V_{CCQH} - 0.01$	 — —	 — —	 V V
Output low voltage[8] • TTL (I_{OL} = 3.0 mA)[6] • CMOS (I_{OL} = 10 µA)[6]	V_{OL}	 — —	 — —	 0.4 0.01	 V V

Table 2-7. Reset, Stop, Mode Select, and Interrupt Timing[5] (CONTINUED)

No.	Characteristics	Expression	200 MHz		220 MHz		240 MHz		275 MHz		Unit
			Min	Max	Min	Max	Min	Max	Min	Max	
20	Delay from $\overline{RD}$ assertion to interrupt request deassertion for level sensitive fast interrupts[1, 6, 7]	(WS + 3.25) × T_C – 10.94	—	Note 7	—	Note 7	—	Note 7	—	Note 7	ns
21	Delay from $\overline{WR}$ assertion to interrupt request deassertion for level sensitive fast interrupts[1, 6, 7] • SRAM WS = 3 • SRAM WS ≥ 4	 (WS + 3) × T_C – 10.94 (WS + 2.5) × T_C – 10.94	 — —	 Note 7 Note 7	 — —	 Note 7 Note 7	 — —	 Note 7 Note 7	 — —	 Note 7 Note 7	 ns ns
24	Duration for $\overline{IRQA}$ assertion to recover from Stop state		8.0	—	8.0	—	8.0	—	8.0	—	ns
25	Delay from $\overline{IRQA}$ assertion to fetch of first instruction (when exiting Stop)[2, 3] • DPLL is not active during Stop (PCTL Bit 1 = 0) and Stop delay is enabled (Operating Mode Register Bit 6 = 0) • DPLL is not active during Stop (PCTL Bit 1 = 0) and Stop delay is not enabled (Operating Mode Register Bit 6 = 1) • DPLL is active during Stop (PCTL Bit 1 = 1; Implies No Stop Delay)	 DPLT + (128K × T_C) DPLT + (23.75 ± 0.5) × T_C (10.0 ± 1.75) × T_C	 662.2 µs 6.9 41.25	 209.9 ms 188.8 58.8	 662.2 µs 6.9 37.5	 209.9 ms 188.8 53.3	 662.2 µs 6.9 34.4	 209.9 ms 188.8 49.0	 662.2 µs 6.9 30.0	 209.9 ms 188.8 43.0	 — µs ns
26	Duration of level sensitive $\overline{IRQA}$ assertion to ensure interrupt service (when exiting Stop)[2, 3] • DPLL is not active during Stop (PCTL bit 1 = 0) and Stop delay is enabled (Operating Mode Register Bit 6 = 0) • DPLL is not active during Stop (PCTL bit 1 = 0) and Stop delay is not enabled (Operating Mode Register Bit 6 = 1) • DPLL is active during Stop ((PCTL bit 1 = 0; implies no Stop delay)	 DPLT + (128 K × T_C) DPLT + (20.5 ± 0.5) × T_C 5.5 × T_C	 805.4 150.1 27.5	 — — —	 805.4 150.1 25	 — — —	 805.4 150.1 22.9	 — — —	 805.4 150.1 20.0	 — — —	 µs µs ns
27	Interrupt Request Rate • HI08, ESSI, SCI, Timer • DMA • $\overline{IRQ}$, $\overline{NMI}$ (edge trigger) • $\overline{IRQ}$, $\overline{NMI}$ (level trigger)	 12T_C 8T_C 8T_C 12T_C	 — — — —	 60.0 40.0 40.0 60.0	 — — — —	 54.6 36.4 36.4 54.6	 — — — —	 50.0 33.4 33.4 50.0	 — — — —	 43.7 29.2 29.2 43.7	 ns ns ns ns
28	DMA Request Rate • Data read from HI08, ESSI, SCI • Data write to HI08, ESSI, SCI • Timer • $\overline{IRQ}$, $\overline{NMI}$ (edge trigger)	 6T_C 7T_C 2T_C 3T_C	 — — — —	 30.0 35.0 10.0 15.0	 — — — —	 27.3 31.9 9.1 13.7	 — — — —	 25.0 29.2 8.3 12.5	 — — — —	 21.84 25.48 7.28 10.92	 ns ns ns ns
29	Delay from $\overline{IRQA}$, $\overline{IRQB}$, $\overline{IRQC}$, $\overline{IRQD}$, $\overline{NMI}$ assertion to external memory (DMA source) access address out valid	4.25 × T_C + 2.0	23.25	—	21.34	—	19.72	—	17.45	—	ns



a) First Interrupt Instruction Execution



b) General-Purpose I/O

Figure 2-4. External Fast Interrupt Timing

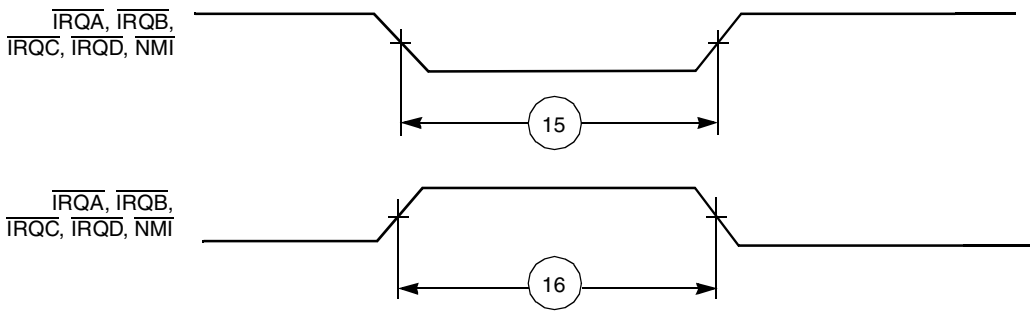
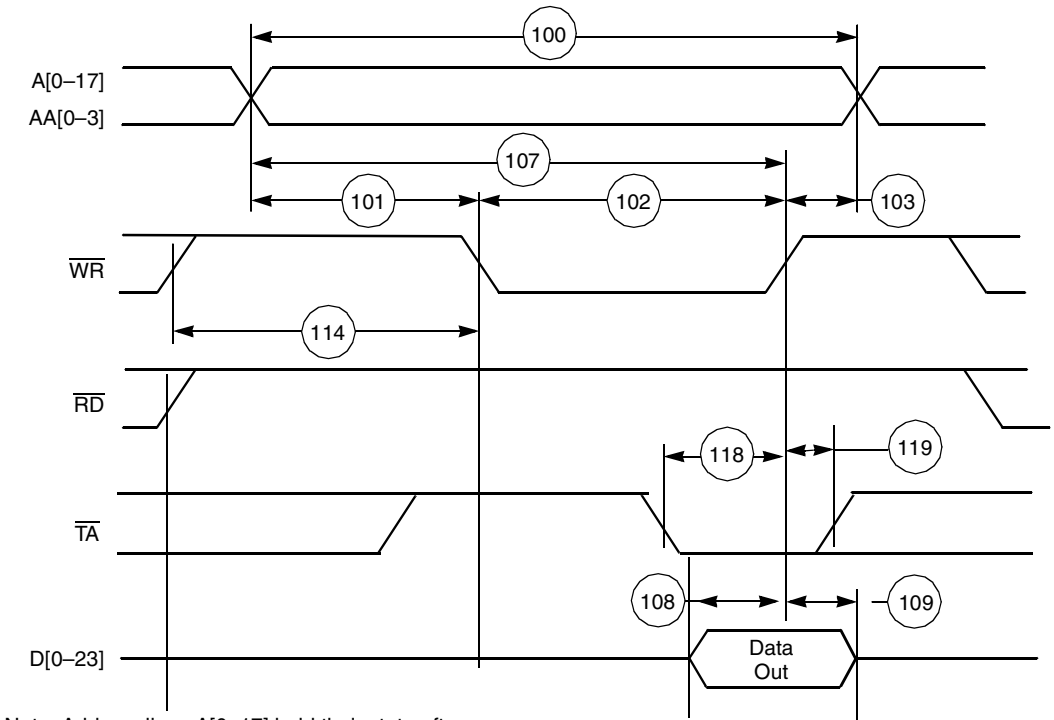


Figure 2-5. External Interrupt Timing (Negative Edge-Triggered)



Note: Address lines A[0–17] hold their state after a read or write operation. AA[0–3] do not hold their state after a read or write operation.

Figure 2-11. SRAM Write Access

2.4.5.2 Asynchronous Bus Arbitration Timings

Table 2-9. Asynchronous Bus Timings

No.	Characteristics	Expression	200 MHz		220 MHz		240 MHz		275 Mhz		Unit
			Min	Max	Min	Max	Min	Max	Min	Max	
250	$\overline{BB}$ assertion window from $\overline{BG}$ input deassertion.	2.5 × Tc + 5	—	17.5	—	16.4	—	15.4	—	14.1	ns
251	Delay from $\overline{BB}$ assertion to $\overline{BG}$ assertion	2 × Tc + 5	15	—	14.1	—	13.3	—	12.27	—	ns

Notes:
1. Bit 13 in the Operating Mode Register must be set to enable Asynchronous Arbitration mode.
2. To guarantee timings 250 and 251, it is recommended that you assert non-overlapping $\overline{BG}$ inputs to different DSP56300 devices (on the same bus), as shown in **Figure 2-12**, where $\overline{BG1}$ is the $\overline{BG}$ signal for one DSP56300 device while $\overline{BG2}$ is the $\overline{BG}$ signal for a second DSP56300 device.

Table 2-10. Host Interface Timings[1,2,12] (Continued)

No.	Characteristic[10]	Expression	200 MHz		220 MHz		240 MHz		275 MHz		Unit
			Min	Max	Min	Max	Min	Max	Min	Max	

Notes:
1. See the Programmer's Model section in the chapter on the HI08 in the *DSP56321 Reference Manual*.
2. In the timing diagrams below, the controls pins are drawn as active low. The pin polarity is programmable.
3. This timing is applicable only if two consecutive reads from one of these registers are executed.
4. The data strobe is Host Read (HRD) or Host Write (HWR) in the Dual Data Strobe mode and Host Data Strobe (HDS) in the Single Data Strobe mode.
5. The read data strobe is HRD in the Dual Data Strobe mode and HDS in the Single Data Strobe mode.
6. The write data strobe is HWR in the Dual Data Strobe mode and HDS in the Single Data Strobe mode.
7. The host request is HREQ in the Single Host Request mode and HRRQ and HTRQ in the Double Host Request mode.
8. The "Last Data Register" is the register at address $7, which is the last location to be read or written in data transfers. This is RXL/TXL in the Big Endian mode (HLEND = 0; HLEND is the Interface Control Register bit 7—ICR[7]), or RXH/TXH in the Little Endian mode (HLEND = 1).
9. In this calculation, the host request signal is pulled up by a 4.7 kΩ resistor in the Open-drain mode.
10. V_{CCQH} = 3.3 V ± 0.3 V, V_{CCQL} = 1.6 V ± 0.1 V; T_J = –40°C to +100 °C, C_L = 50 pF
11. This timing is applicable only if a read from the "Last Data Register" is followed by a read from the RXL, RXM, or RXH registers without first polling RXDF or HREQ bits, or waiting for the assertion of the HREQ signal.
12. After the external host writes a new value to the ICR, the HI08 will be ready for operation after three DSP clock cycles (3 × Tc).

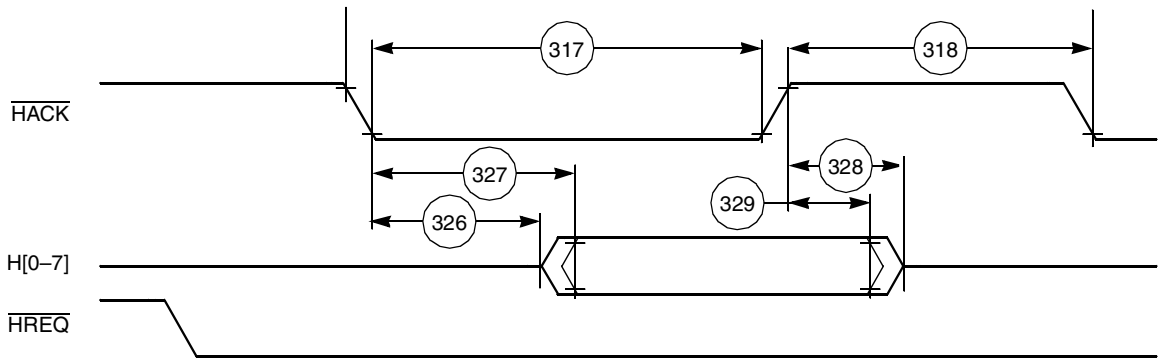


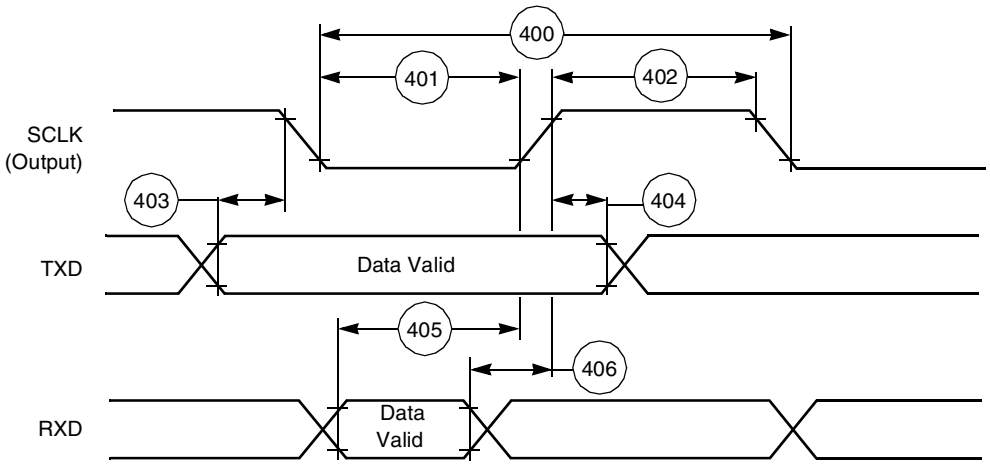
Figure 2-13. Host Interrupt Vector Register (IVR) Read Timing Diagram

2.4.7 SCI Timing

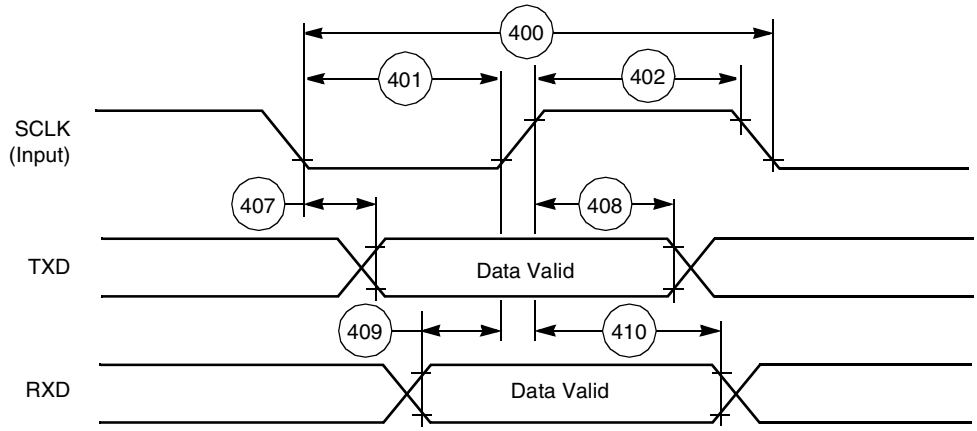
Table 2-11. SCI Timings

No.	Characteristics[1]	Symbol	Expression	200 MHz		220 MHz		240 MHz		275 MHz		Unit
				Min	Max	Min	Max	Min	Max	Min	Max	
400	Synchronous clock cycle	t_{SCC}[2]	$16 \times T_C$	80.0	—	72.8	—	66.7	—	58.0	—	ns
401	Clock low period		$t_{SCC}/2 - 10.0$	30.0	—	26.4	—	23.4	—	19.0	—	ns
402	Clock high period		$t_{SCC}/2 - 10.0$	30.0	—	26.4	—	23.4	—	19.0	—	ns
403	Output data setup to clock falling edge (internal clock)		$t_{SCC}/4 + 0.5 \times T_C - 17.0$	5.5	—	3.5	—	1.76	—	–0.68	—	ns
404	Output data hold after clock rising edge (internal clock)		$t_{SCC}/4 - 1.5 \times T_C$	13	—	11.5	—	10	—	9.04	—	ns
405	Input data setup time before clock rising edge (internal clock)		$t_{SCC}/4 + 0.5 \times T_C + 25.0$	47.5	—	45.5	—	43.8	—	41.32	—	ns
406	Input data not valid before clock rising edge (internal clock)		$t_{SCC}/4 + 0.5 \times T_C - 5.5$	—	17.0	—	15.0	—	13.8	—	10.81	ns
407	Clock falling edge to output data valid (external clock)			—	32.0	—	32.0	—	32.0	—	32.0	ns
408	Output data hold after clock rising edge (external clock)		$T_C + 8.0$	13.0	—	12.6	—	12.2	—	11.64	—	ns
409	Input data setup time before clock rising edge (external clock)			0.0	—	0.0	—	0.0	—	0.0	—	ns
410	Input data hold time after clock rising edge (external clock)			9.0	—	9.0	—	9.0	—	9.0	—	ns
411	Asynchronous clock cycle	t_{ACC}[3]	$64 \times T_C$	320.0	—	291.2	—	266.9	—	232.0	—	ns
412	Clock low period		$t_{ACC}/2 - 10.0$	150.0	—	135.6	—	123.5	—	106.0	—	ns
413	Clock high period		$t_{ACC}/2 - 10.0$	150.0	—	135.6	—	123.5	—	106.0	—	ns
414	Output data setup to clock rising edge (internal clock)		$t_{ACC}/2 - 30.0$	130.0	—	115.6	—	103.5	—	86.0	—	ns
415	Output data hold after clock rising edge (internal clock)		$t_{ACC}/2 - 30.0$	130.0	—	115.6	—	103.5	—	86.0	—	ns

Notes:
1. $V_{CCQH} = 3.3\ V \pm 0.3\ V$, $V_{CCQL} = 1.6\ V \pm 0.1\ V$; $T_J = -40°C$ to $+100\ °C$, $C_L = 50\ pF$.
2. t_{SCC} = synchronous clock cycle time (for internal clock, t_{SCC} is determined by the SCI clock control register and T_C).
3. t_{ACC} = asynchronous clock cycle time; value given for 1X Clock mode (for internal clock, t_{ACC} is determined by the SCI clock control register and T_C).
4. In the timing diagrams that follow, the SCLK is drawn using the clock falling edge as a the first reference. Clock polarity is programmable in the SCI Control Register (SCR). Refer to the *DSP56321 Reference Manual* for details.



a) Internal Clock



b) External Clock

Figure 2-22. SCI Synchronous Mode Timing

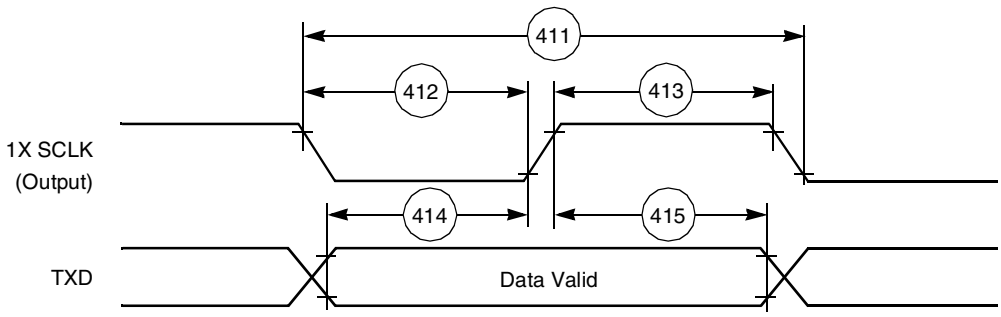


Figure 2-23. SCI Asynchronous Mode Timing

Packaging 3

This section includes diagrams of the DSP56321 package pin-outs and tables showing how the signals described in **Chapter 1** are allocated for the package. The DSP56321 is available in a 196-pin molded array plastic-ball grid array (MAP-BGA) package.

Table 3-2. Signal List by Signal Name (Continued)

Signal Name	Ball No.	Signal Name	Ball No.	Signal Name	Ball No.
MODA	C4	PB4	N3	Reserved	M9
MODB	A5	PB5	P2	Reserved	N10
MODC	C5	PB6	N1	$\overline{\text{RESET}}$	N5
MODD	B5	PB7	N2	RXD	F1
NC	A1	PB8	M3	SC00	F3
NC	A14	PB9	M1	SC01	D2
NC	B14	PC0	F3	SC02	C1
NC	M10	PC1	D2	SC10	F2
NC	N8	PC2	C1	SC11	A2
NC	P1	PC3	H3	SC12	B2
NC	P5	PC4	E3	SCK0	H3
NC	P14	PC5	E1	SCK1	G1
$\overline{\text{NMI}}$	D1	PD0	F2	SCLK	G2
PB0	M5	PD1	A2	SRD0	E3
PB1	P4	PD2	B2	SRD1	B1
PB10	M2	PD3	G1	STD0	E1
PB11	J2	PD4	B1	STD1	C2
PB12	J3	PD5	C2	$\overline{\text{TA}}$	P10
PB13	L1	PE0	F1	TCK	C3
PB14	K2	PE1	G3	TDI	B3
PB15	J1	PE2	G2	TDO	A4
PB2	N4	PINIT	D1	TIO0	L3
PB3	P3	$\overline{\text{RD}}$	M12	TIO1	L2

Design Considerations 4

This section describes various areas to consider when incorporating the DSP56321 device into a system design.

4.1 Thermal Design Considerations

An estimate of the chip junction temperature, T_J, in °C can be obtained from this equation:

Equation 1: $$T_J = T_A + (P_D \times R_{\theta JA})$$

Where:

T_A	=	ambient temperature °C
$R_{\theta JA}$	=	package junction-to-ambient thermal resistance °C/W
P_D	=	power dissipation in package

Historically, thermal resistance has been expressed as the sum of a junction-to-case thermal resistance and a case-to-ambient thermal resistance, as in this equation:

Equation 2: $$R_{\theta JA} = R_{\theta JC} + R_{\theta CA}$$

Where:

$R_{\theta JA}$	=	package junction-to-ambient thermal resistance °C/W
$R_{\theta JC}$	=	package junction-to-case thermal resistance °C/W
$R_{\theta CA}$	=	package case-to-ambient thermal resistance °C/W

$R_{\theta JC}$ is device-related and cannot be influenced by the user. The user controls the thermal environment to change the case-to-ambient thermal resistance, $R_{\theta CA}$. For example, the user can change the air flow around the device, add a heat sink, change the mounting arrangement on the printed circuit board (PCB) or otherwise change the thermal dissipation capability of the area surrounding the device on a PCB. This model is most useful for ceramic packages with heat sinks; some 90 percent of the heat flow is dissipated through the case to the heat sink and out to the ambient environment. For ceramic packages, in situations where the heat flow is split between a path to the case and an alternate path through the PCB, analysis of the device thermal performance may need the additional modeling capability of a system-level thermal simulation tool.

The thermal performance of plastic packages is more dependent on the temperature of the PCB to which the package is mounted. Again, if the estimates obtained from $R_{\theta JA}$ do not satisfactorily answer whether the thermal performance is adequate, a system-level model may be appropriate.

A complicating factor is the existence of three common ways to determine the junction-to-case thermal resistance in plastic packages.

- To minimize temperature variation across the surface, the thermal resistance is measured from the junction to the outside surface of the package (case) closest to the chip mounting area when that surface has a proper heat sink.

- Consider all device loads as well as parasitic capacitance due to PCB traces when you calculate capacitance. This is especially critical in systems with higher capacitive loads that could create higher transient currents in the V_{CC} and GND circuits.
- All inputs must be terminated (that is, not allowed to float) by CMOS levels except for the three pins with internal pull-up resistors ($\overline{\text{TRST}}$, TMS, $\overline{\text{DE}}$).
- The following pins must be asserted during the power-up sequence: $\overline{\text{RESET}}$ and $\overline{\text{TRST}}$. A stable EXTAL signal should be supplied before deassertion of $\overline{\text{RESET}}$. If the V_{CC} reaches the required level before EXTAL is stable or other "required $\overline{\text{RESET}}$ duration" conditions are met (see **Table 2-7**), the device circuitry can be in an uninitialized state that may result in significant power consumption and heat-up. Designs should minimize this condition to the shortest possible duration.
- Ensure that during power-up, and throughout the DSP56321 operation, V_{CCQH} is always higher or equal to the V_{CCQL} voltage level.
- If multiple DSP devices are on the same board, check for cross-talk or excessive spikes on the supplies due to synchronous operation of the devices.
- The Port A data bus (D[0–23]), HI08, ESSI0, ESSI1, SCI, and timers all use internal keepers to maintain the last output value even when the internal signal is tri-stated. Typically, no pull-up or pull-down resistors should be used with these signal lines. However, if the DSP is connected to a device that requires pull-up resistors (such as an MPC8260), the recommended resistor value is 10 KΩ or less. If more than one DSP must be connected in parallel to the other device, the pull-up resistor value requirement changes as follows:
  - 2 DSPs = 5 KΩ (mask sets 0K91M and 1K91M)/7 KΩ (mask set 0K93M) or less
  - 3 DSPs = 3 KΩ (mask sets 0K91M and 1K91M)/4 KΩ (mask set 0K93M) or less
  - 4 DSPs = 2 KΩ (mask sets 0K91M and 1K91M)/3 KΩ (mask set 0K93M) or less
  - 5 DSPs = 1.5 KΩ (mask sets 0K91M and 1K91M)/2 KΩ (mask set 0K93M) or less
  - 6 DSPs = 1 KΩ (mask sets 0K91M and 1K91M)/1.5 KΩ (mask set 0K93M) or less

Note: Refer to *EB610/D DSP56321/DSP56321T Power-Up Sequencing Guidelines* for detailed information about minimizing power consumption during startup.

4.3 Power Consumption Considerations

Power dissipation is a key issue in portable DSP applications. Some of the factors affecting current consumption are described in this section. Most of the current consumed by CMOS devices is alternating current (ac), which is charging and discharging the capacitances of the pins and internal nodes.

Current consumption is described by this formula:

Equation 3: $I = C \times V \times f$

Where:

C	=	node/pin capacitance
V	=	voltage swing
f	=	frequency of node/pin toggle

Example 4-1. Current Consumption

For a Port A address pin loaded with 50 pF capacitance, operating at 3.3 V, with a 66 MHz clock, toggling at its maximum possible rate (33 MHz), the current consumption is expressed in **Equation 4**.

Power Consumption Benchmark A

The following benchmark program evaluates DSP56321 power use in a test situation. It enables the PLL, disables the external clock, and uses repeated multiply-accumulate (MAC) instructions with a set of synthetic DSP application data to emulate intensive sustained DSP operation.

```
;*************************************************************************
;*************************************************************************
;*                                                                       *
;*                    CHECKS   Typical Power Consumption                 *
;*                                                                       *
;*************************************************************************

       page   200,55,0,0,0
       nolist

I_VEC EQU $000000; Interrupt vectors for program debug only
START EQU $8000; MAIN (external) program starting address
INT_PROG  EQU $100 ; INTERNAL program memory starting address
INT_XDAT  EQU $0; INTERNAL X-data memory starting address
INT_YDAT  EQU $0; INTERNAL Y-data memory starting address

       INCLUDE "ioequ.asm"
       INCLUDE "intequ.asm"

       list

       org    P:START
;
       movep #$0243FF,x:M_BCR ;   ; BCR: Area 3 = 2 w.s (SRAM)
; Default: 2w.s (SRAM)
;
       movep  #$00000F,x:M_PCTL   ; XTAL disable
                                  ; PLL enable
;
; Load the program
;
       move   #INT_PROG,r0
       move   #PROG_START,r1
       do     #(PROG_END-PROG_START),PLOAD_LOOP
       move   p:(r1)+,x0
       move   x0,p:(r0)+
       nop
PLOAD_LOOP
;
; Load the X-data
;
       move   #INT_XDAT,r0
       move   #XDAT_START,r1
       do     #(XDAT_END-XDAT_START),XLOAD_LOOP
       move   p:(r1)+,x0
       move   x0,x:(r0)+
XLOAD_LOOP
```

```
;
; Load the Y-data
;
       move   #INT_YDAT,r0
       move   #YDAT_START,r1
       do     #(YDAT_END-YDAT_START),YLOAD_LOOP
       move   p:(r1)+,x0
       move   x0,y:(r0)+
YLOAD_LOOP
;

       jmp    INT_PROG

PROG_START
       move   #$0,r0
       move   #$0,r4
       move   #$3f,m0
       move   #$3f,m4
;
       clr    a
       clr    b
       move   #$0,x0
       move   #$0,x1
       move   #$0,y0
       move   #$0,y1
       bset   #4,omr        ; ebd
;
sbr    dor    #60,_end
       mac    x0,y0,ax:(r0)+,x1   y:(r4)+,y1
       mac    x1,y1,ax:(r0)+,x0   y:(r4)+,y0
       add    a,b
       mac    x0,y0,ax:(r0)+,x1
       mac    x1,y1,a             y:(r4)+,y0
       move   b1,x:$ff
_end
       bra    sbr
       nop
       nop
       nop
       nop
PROG_END
       nop
       nop


XDAT_START
;      org    x:0
       dc     $262EB9
       dc     $86F2FE
       dc     $E56A5F
       dc     $616CAC
       dc     $8FFD75
       dc     $9210A
       dc     $A06D7B
       dc     $CEA798
       dc     $8DFBF1
       dc     $A063D6
       dc     $6C6657
       dc     $C2A544
       dc     $A3662D
       dc     $A4E762
       dc     $84F0F3
       dc     $E6F1B0
```

```
ioequ   ident   1,0

;---------------------------------------------------------------------------
;
;       EQUATES for I/O Port Programming
;
;---------------------------------------------------------------------------

;       Register Addresses

M_HDR EQU $FFFFC9                ; Host port GPIO data Register
M_HDDR EQU $FFFFC8               ; Host port GPIO direction Register
M_PCRC EQU $FFFFBF               ; Port C Control Register
M_PRRC EQU $FFFFBE               ; Port C Direction Register
M_PDRC EQU $FFFFBD               ; Port C GPIO Data Register
M_PCRD EQU $FFFFAF               ; Port D Control register
M_PRRD EQU $FFFFAE               ; Port D Direction Data Register
M_PDRD EQU $FFFFAD               ; Port D GPIO Data Register
M_PCRE EQU $FFFF9F               ; Port E Control register
M_PRRE EQU $FFFF9E               ; Port E Direction Register
M_PDRE EQU $FFFF9D               ; Port E Data Register
M_OGDB EQU $FFFFFC               ; OnCE GDB Register


;---------------------------------------------------------------------------
;
;       EQUATES for Host Interface
;
;---------------------------------------------------------------------------

;       Register Addresses

M_HCR EQU $FFFFC2                ; Host Control Register
M_HSR EQU $FFFFC3                ; Host Status Register
M_HPCR EQU $FFFFC4               ; Host Polarity Control Register
M_HBAR EQU $FFFFC5               ; Host Base Address Register
M_HRX EQU $FFFFC6                ; Host Receive Register
M_HTX EQU $FFFFC7                ; Host Transmit Register

;       HCR bits definition
M_HRIE EQU $0                    ; Host Receive interrupts Enable
M_HTIE EQU $1                    ; Host Transmit Interrupt Enable
M_HCIE EQU $2                    ; Host Command Interrupt Enable
M_HF2 EQU $3                     ; Host Flag 2
M_HF3 EQU $4                     ; Host Flag 3

;       HSR bits definition
M_HRDF EQU $0                    ; Host Receive Data Full
M_HTDE EQU $1                    ; Host Receive Data Empty
M_HCP EQU $2                     ; Host Command Pending
M_HF0 EQU $3                     ; Host Flag 0
M_HF1 EQU $4                     ; Host Flag 1


;       HPCR bits definition
M_HGEN EQU $0                    ; Host Port GPIO Enable
M_HA8EN EQU $1                   ; Host Address 8 Enable
M_HA9EN EQU $2                   ; Host Address 9 Enable
M_HCSEN EQU $3                   ; Host Chip Select Enable
M_HREN EQU $4                    ; Host Request Enable
M_HAEN EQU $5                    ; Host Acknowledge Enable
M_HEN EQU $6                     ; Host Enable
```

```
M_PCOD    EQU     0               ; PLL Clock Output Disable Bit
M_PSTP    EQU     1               ; STOP Processing State Bit
M_XTLD    EQU     2               ; XTAL Disable Bit
M_PEN     EQU     3               ; PLL Enable Bit

;------------------------------------------------------------------------
;
;       EQUATES for BIU
;
;------------------------------------------------------------------------

;       Register Addresses Of BIU


M_BCR EQU $FFFFFB               ; Bus Control Register
M_DCR EQU $FFFFFA               ; DRAM Control Register
M_AAR0 EQU $FFFFF9              ; Address Attribute Register 0
M_AAR1 EQU $FFFFF8              ; Address Attribute Register 1
M_AAR2 EQU $FFFFF7              ; Address Attribute Register 2
M_AAR3 EQU $FFFFF6              ; Address Attribute Register 3
M_IDR EQU  $FFFFF5              ; ID Register

;       Bus Control Register

M_BA0W EQU $1F                  ; Area 0 Wait Control Mask (BA0W0-BA0W4)
M_BA1W EQU $3E0                 ; Area 1 Wait Control Mask (BA1W0-BA14)
M_BA2W EQU $1C00                ; Area 2 Wait Control Mask (BA2W0-BA2W2)
M_BA3W EQU $E000                ; Area 3 Wait Control Mask (BA3W0-BA3W3)
M_BDFW EQU $1F0000              ; Default Area Wait Control Mask (BDFW0-BDFW4)
M_BBS EQU 21                    ; Bus State
M_BLH EQU 22                    ; Bus Lock Hold
M_BRH EQU 23                    ; Bus Request Hold

;       DRAM Control Register

M_BCW EQU $3                    ; In Page Wait States Bits Mask (BCW0-BCW1)
M_BRW EQU $C                    ; Out Of Page Wait States Bits Mask (BRW0-BRW1)
M_BPS EQU $300                  ; DRAM Page Size Bits Mask (BPS0-BPS1)
M_BPLE EQU 11                   ; Page Logic Enable
M_BME EQU 12                    ; Mastership Enable
M_BRE EQU 13                    ; Refresh Enable
M_BSTR EQU 14                   ; Software Triggered Refresh
M_BRF EQU $7F8000               ; Refresh Rate Bits Mask (BRF0-BRF7)
M_BRP EQU 23                    ; Refresh prescaler

;       Address Attribute Registers

M_BAT EQU $3                    ; Ext. Access Type and Pin Def. Bits Mask (BAT0-BAT1)
M_BAAP EQU 2                    ; Address Attribute Pin Polarity
M_BPEN EQU 3                    ; Program Space Enable
M_BXEN EQU 4                    ; X Data Space Enable
M_BYEN EQU 5                    ; Y Data Space Enable
M_BAM EQU 6                     ; Address Muxing
M_BPAC EQU 7                    ; Packing Enable
M_BNC EQU $F00                  ; Number of Address Bits to Compare Mask (BNC0-BNC3)
M_BAC EQU $FFF000               ; Address to Compare Bits Mask (BAC0-BAC11)


;       control and status bits in SR

M_CP EQU $c00000                ; mask for CORE-DMA priority bits in SR
M_CA EQU 0                      ; Carry
M_V EQU 1                       ; Overflow
```

```
;---------------------------------------------------------------------
; Non-Maskable interrupts
;---------------------------------------------------------------------
I_RESET EQU I_VEC+$00             ; Hardware RESET
I_STACK EQU I_VEC+$02             ; Stack Error
I_ILL EQU I_VEC+$04               ; Illegal Instruction
I_DBG EQU I_VEC+$06               ; Debug Request
I_TRAP EQU I_VEC+$08              ; Trap
I_NMI EQU I_VEC+$0A               ; Non Maskable Interrupt

;---------------------------------------------------------------------
; Interrupt Request Pins
;---------------------------------------------------------------------
I_IRQA EQU I_VEC+$10              ; IRQA
I_IRQB EQU I_VEC+$12              ; IRQB
I_IRQC EQU I_VEC+$14              ; IRQC
I_IRQD EQU I_VEC+$16              ; IRQD

;---------------------------------------------------------------------
; DMA Interrupts
;---------------------------------------------------------------------
I_DMA0 EQU I_VEC+$18              ; DMA Channel 0
I_DMA1 EQU I_VEC+$1A              ; DMA Channel 1
I_DMA2 EQU I_VEC+$1C              ; DMA Channel 2
I_DMA3 EQU I_VEC+$1E              ; DMA Channel 3
I_DMA4 EQU I_VEC+$20              ; DMA Channel 4
I_DMA5 EQU I_VEC+$22              ; DMA Channel 5

;---------------------------------------------------------------------
; Timer Interrupts
;---------------------------------------------------------------------
I_TIM0C EQU I_VEC+$24             ; TIMER 0 compare
I_TIM0OF EQU I_VEC+$26            ; TIMER 0 overflow
I_TIM1C EQU I_VEC+$28             ; TIMER 1 compare
I_TIM1OF EQU I_VEC+$2A            ; TIMER 1 overflow
I_TIM2C EQU I_VEC+$2C             ; TIMER 2 compare
I_TIM2OF EQU I_VEC+$2E            ; TIMER 2 overflow

;---------------------------------------------------------------------
; ESSI Interrupts
;---------------------------------------------------------------------
I_SI0RD EQU I_VEC+$30             ; ESSI0 Receive Data
I_SI0RDE EQU I_VEC+$32            ; ESSI0 Receive Data w/ exception Status
I_SI0RLS EQU I_VEC+$34            ; ESSI0 Receive last slot
I_SI0TD EQU I_VEC+$36             ; ESSI0 Transmit data
I_SI0TDE EQU I_VEC+$38            ; ESSI0 Transmit Data w/ exception Status
I_SI0TLS EQU I_VEC+$3A            ; ESSI0 Transmit last slot
I_SI1RD EQU I_VEC+$40             ; ESSI1 Receive Data
I_SI1RDE EQU I_VEC+$42            ; ESSI1 Receive Data w/ exception Status
I_SI1RLS EQU I_VEC+$44            ; ESSI1 Receive last slot
I_SI1TD EQU I_VEC+$46             ; ESSI1 Transmit data
I_SI1TDE EQU I_VEC+$48            ; ESSI1 Transmit Data w/ exception Status
I_SI1TLS EQU I_VEC+$4A            ; ESSI1 Transmit last slot

;---------------------------------------------------------------------
; SCI Interrupts
;---------------------------------------------------------------------
I_SCIRD EQU I_VEC+$50             ; SCI Receive Data
I_SCIRDE EQU I_VEC+$52            ; SCI Receive Data With Exception Status
I_SCITD EQU I_VEC+$54             ; SCI Transmit Data
I_SCIIL EQU I_VEC+$56             ; SCI Idle Line
I_SCITM EQU I_VEC+$58             ; SCI Timer
```