

Chapter 11

Operating Modes and Memory Spaces

The DSP56300 family core mode pins (MODA, MODB, MODC, and MODD) determine the reset vector address that points to the start-up procedure when the device leaves the Reset state. The mode pins are sampled as the device exits from Reset. The sampled state of these pins is subject to a mask-programmed look-up table that can be used as a filter to disable the user from entering some of the operating modes. This filtered state is written to the MD, MC, MB, and MA bits in the Operating Mode Register (OMR). When the Reset state is exited, the mode pins become general-purpose interrupt pins, \overline{IRQA} , \overline{IRQB} , \overline{IRQC} , and \overline{IRQD} . When the device is not in the Reset state, software can change the OMR mode bits (MA, MB, MC, and MD). **Table 11-1** lists the mode assignments in the DSP56300 family core. The reset vector is chosen from device-specific addresses: RESET1, RESET2, and RESET3. Each reset vector in a specific DSP56300 family device is assigned one of two different values. **Table 11-2** shows typical values. These reset vectors are implementation-specific.

Table 11-1. DSP Core Operating Modes

MOD[D:A]	Mode	Description	Reset Vector
0000	0	Expanded Mode 0	RESET1
0001–0111	1–7	System Configuration Mode 1–7	RESET3
1000	8	Expanded Mode 8	RESET2
1001–1111	9–F	System Configuration Mode 9–F	RESET3

Table 11-2. DSP Core Reset Vectors, Possible Values

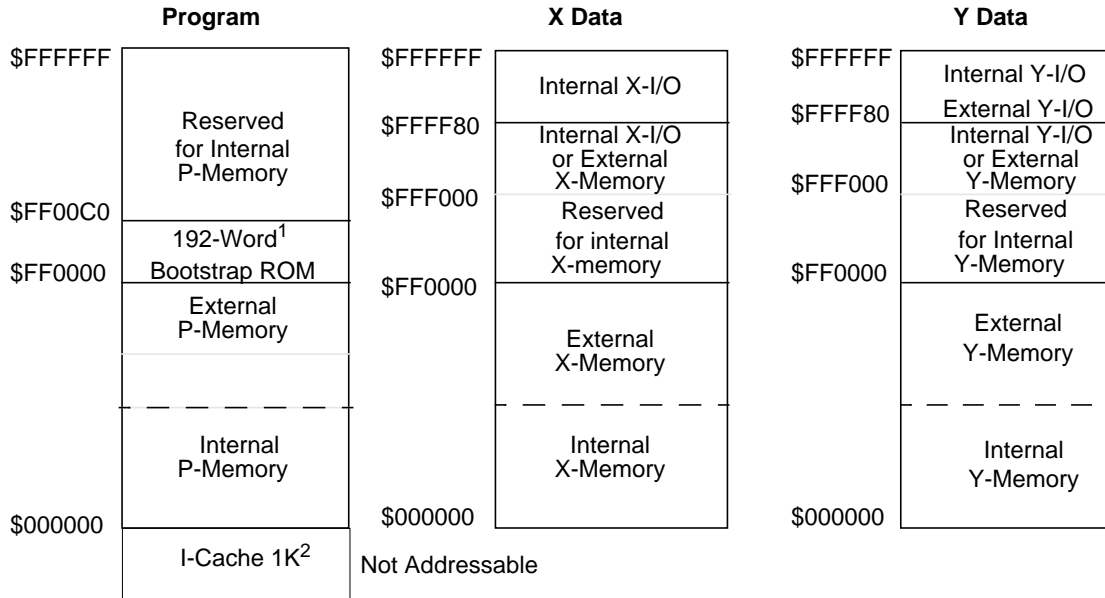
RESET1	RESET2	RESET3
\$000000	\$004000	\$000000
\$C00000	\$008000	\$FF0000

In Expanded Modes 0 and 8, a hardware reset causes the DSP56300 family core to jump to the mask-programmed external program memory location RESET1 or RESET2, respectively, and execute the code fetched from this location. These locations are implementation specific. See the appropriate user’s manual for more information.

In the System Configuration Modes 1–7 and 9–F, a hardware reset causes the DSP56300 family core to jump to the mask-programmed internal program memory (usually ROM) location RESET3, and execute the code fetched from this location. These routines are typically implementation-specific, and can be contained in the bootstrap code.

11.1 DSP56300 Family Core Memory Map

The memory space of the DSP56300 family core is partitioned into program memory space (P), X data memory space, and Y data memory space. The data memory space is divided into X data memory and Y data memory in order to work with the two Address Arithmetic Logic Units (Address ALUs) and to feed two operands simultaneously to the Data ALU. Each memory space may include internal RAM, and/or internal ROM and can be expanded off-chip under software control. **Figure 11-1** shows the three independent memory spaces of the DSP56300 family core: X data, Y data, and program.



NOTE 1: In recent revisions of some DSP56300 family members, the size of the Bootstrap ROM is 3K, so the Bootstrap ROM size measures \$FF0000 – \$FF0C00.

NOTE 2: External program memory begins immediately after the internal program memory. The internal memory modules that are mapped to the addresses up to \$00C00 – \$001000 are used as I-Cache space when the I-Cache is enabled, and these addresses become part of the external P memory space.

Figure 11-1. DSP56300 Core Memory Map

Note: Individual members of the DSP56300 family can have different amounts of X data, Y data, and program memory. Consult the appropriate user's manual and technical data sheet for more information.

11.1.1 X Data Memory Space

The X data memory space is divided into five parts:

- Internal X I/O space
- Switchable internal or external X I/O memory space
- Reserved space for X ROM or RAM
- External X data memory
- Internal X data RAM

11.1.2 Internal X I/O Space

The on-chip X I/O peripheral registers occupy the top 128 locations of the X data memory space (\$FF80 – \$FFFF) and can be accessed by the MOVE and MOVEP instructions, as well as by bit-oriented instructions, such as the BCHG, BCLR, BSET, BTST, BRCLR, BRSET, BSCLR, BSSET, JCLR, JSET, JSCLR, and JSSET. Some of the DSP56300 family core registers are mapped to the internal X I/O space as well, as **Table 11-3** shows.

Table 11-3. Internal X I/O Space Map

Register	Block	Address	Register Name and Description
IIRC	PIC	\$FFFFFFF	Interrupt Priority Register Core
IIRP		\$FFFFFFE	Interrupt Priority Register Peripheral
PCTL	PLL	\$FFFFFFD	PLL Control Register
OGDB	OnCE	\$FFFFFFC	OnCE GDB Register
BCR	PORT A	\$FFFFFFB	Bus Control Register
DCR		\$FFFFFFA	DRAM Control Register
AAR0		\$FFFFFF9	Address Attribute Register 0
AAR1		\$FFFFFF8	Address Attribute Register 1
AAR2		\$FFFFFF7	Address Attribute Register 2
AAR3		\$FFFFFF6	Address Attribute Register 3
IDR		\$FFFFFF5	ID Register

Table 11-3. Internal X I/O Space Map (Continued)

Register	Block	Address	Register Name and Description
DSTR	DMA	\$FFFFFF4	DMA Status Register
DOR0		\$FFFFFF3	DMA Offset Register 0
DOR1		\$FFFFFF2	DMA Offset Register 1
DOR2		\$FFFFFF1	DMA Offset Register 2
DOR3		\$FFFFFF0	DMA Offset Register 3
DSR0	DMA Channel 0	\$FFFFEF	DMA Source Address Register
DDR0		\$FFFFEE	DMA Destination Address Register
DCO0		\$FFFFED	DMA Counter
DCR0		\$FFFFEC	DMA Control Register
DSR1	DMA Channel 1	\$FFFFEB	DMA Source Address Register
DDR1		\$FFFFEA	DMA Destination Address Register
DCO1		\$FFFFE9	DMA Counter
DCR1		\$FFFFE8	DMA Control Register
DSR2	DMA Channel 2	\$FFFFE7	DMA Source Address Register
DDR2		\$FFFFE6	DMA Destination Address Register
DCO2		\$FFFFE5	DMA Counter
DCR2		\$FFFFE4	DMA Control Register
DSR3	DMA Channel 3	\$FFFFE3	DMA Source Address Register
DDR3		\$FFFFE2	DMA Destination Address Register
DCO3		\$FFFFE1	DMA Counter
DCR3		\$FFFFE0	DMA Control Register
DSR4	DMA Channel 4	\$FFFFDF	DMA Source Address Register
DDR4		\$FFFFDE	DMA Destination Address Register
DCO4		\$FFFFDD	DMA Counter
DCR4		\$FFFFDC	DMA Control Register
DSR5	DMA Channel 5	\$FFFFDB	DMA Source Address Register
DDR5		\$FFFFDA	DMA Destination Address Register
DCO5		\$FFFFD9	DMA Counter
DCR5		\$FFFFD8	DMA Control Register

Table 11-3. Internal X I/O Space Map (Continued)

Register	Block	Address	Register Name and Description
Reserved	On-Chip X-I/O mapped Registers	\$FFFFD7	Reserved for On-Chip X-I/O mapped Register
		..	Reserved for On-Chip X-I/O mapped Register
		..	Reserved for On-Chip X-I/O mapped Register
		..	Reserved for On-Chip X-I/O mapped Register
		\$FFFF80	Reserved for On-Chip X- I/O mapped Register

11.1.3 Switchable Internal or External X I/O Memory

The X memory space \$FFF000 – \$FFFF7F is device-specific and is either external X data memory or internal X I/O space for on-chip memory-mapped peripheral registers.

11.1.3.1 Reserved Space for X ROM or RAM

The X memory space \$FF0000 – \$FFEFFF is reserved for inclusion of X data ROM or RAM modules (2048 locations each). The importance of modular organization of the X ROM/RAM becomes apparent in the case of a DMA access to the internal X memory simultaneous with a core access to the same space. DMA and core accesses to different banks can be completed at full speed, while accesses to the same bank halt the DMA until a program memory slot is available.

11.1.3.2 External X Data Memory

The X memory space \$000000 – \$FEFFFF is for expanding to external X memory. The starting address of the external X data memory space is device-dependent. Refer to the appropriate user's manual to determine the actual address used in that device.

11.1.3.3 Internal X Memory

The X memory space \$000000 – \$00FFFF is for internal X RAM modules (256 locations each). The last address of the internal X memory is device-dependent. Refer to the appropriate user's manual to determine the actual address used in that device. The importance of modular organization of the X RAM becomes apparent during a DMA access to the internal X memory simultaneous with a core access to the same space. DMA and core accesses to different banks can be completed at full speed, while accesses to the same bank halt the DMA until a program memory slot is available.

11.1.4 Y Data Memory Space

The Y data memory space is divided into five parts:

- Internal/External Y I/O space
- Switchable internal or external Y I/O memory space
- Reserved space for Y ROM or RAM
- External Y data memory
- Internal Y data RAM

11.1.4.1 Internal/External Y I/O Space

The off-chip or on-chip Y I/O peripheral registers occupy the top 128 locations of the Y data memory space (\$FFFF80 – \$FFFFFF) and can be accessed by MOVE and MOVEP instructions and by bit-oriented instructions (BCHG, BCLR, BSET, BTST, BRCLR, BRSET, BSCLR, BSSET, JCLR, JSET, JSCLR and JSSET). This space is partitioned into eight equal parts (16 locations each). Each part is device-specific and is either external Y I/O or internal Y I/O space.

11.1.4.2 Switchable Internal or External Y I/O Memory

The Y memory space \$FFF000 – \$FFFF7F is device-specific and is either external Y data memory or internal Y I/O space for on-chip memory-mapped peripheral registers.

11.1.4.3 Reserved Space for Y ROM or RAM

The Y memory space \$FF0000 – \$FFEFFF is reserved for inclusion of Y data ROM or RAM modules (2048 locations each). The importance of modular organization of the Y ROM/RAM becomes apparent in the case of a DMA access to the internal Y memory simultaneous with a core access to the same space. DMA and core accesses to different banks can be completed at full speed, while accesses to the same bank halt the DMA until a program memory slot is available.

11.1.4.4 External Y Data Memory

The Y data memory space \$000000 – \$FEFFFF is for expanding to external Y data memory. The starting address of the external Y data memory space is device-dependent. Refer to the appropriate user's manual to determine the actual address used in that device.

11.1.4.5 Internal Y Memory

The Y memory space \$000000 – \$00FFFF is for internal Y RAM modules (256 locations each). The last address of the internal Y memory is device-dependent. Refer to the

appropriate user's manual to determine the actual address used in that device. The importance of modular organization of the Y RAM becomes apparent in the case of a DMA access to the internal Y memory simultaneous with a core access to the same space. DMA and core accesses to different banks can be completed at full speed, while accesses to the same bank halt the DMA until a program memory slot is available.

11.1.5 Program Memory

The Program memory space is divided into five parts:

- Bootstrap ROM (192 words)
- Reserved space for Program ROM
- External program memory
- Internal program memory
- Internal instruction cache memory

11.1.5.1 Bootstrap ROM Space

The program memory space \$FF0000 – \$FF00BF is for the internal bootstrap ROM. The ROM contains 192 words combining the bootstrap program for the specific DSP56300 family device. The bootstrap ROM space cannot be accessed by DMA.

11.1.5.2 Reserved Space for Program ROM

The program memory space \$FF00C0 – \$FFFFFF is reserved for inclusion of Program ROM modules (2048 locations each). Program ROM may be used to contain some operating system program or other application-specific pre-defined user programs. The importance of modular organization of the Program ROM space is apparent in the case of DMA access to the internal program memory simultaneous with core access to the same space. DMA and core accesses to different banks can be completed at full speed, while accesses to the same bank halt the DMA until a program memory slot is available.

11.1.5.3 External Program Memory

The program memory space \$000000 – \$FEFFFF is for expanding to external program memory. The starting address of the external program memory space is device-dependent and also depends on the amount of on-chip Program RAM and the Instruction Cache size. Refer to the appropriate user's manual to determine the actual address used in that device.

11.1.5.4 Internal Program Memory

The program memory space \$000000 – \$00FFFF is for internal Program RAM modules (256 locations for each RAM module). The last address of the internal program memory is

device-dependent. Refer to the appropriate user's manual to determine the actual address used in that device. The importance of modular organization of the program memory becomes apparent in the case of a DMA access to the internal program memory simultaneous with a core access to the same space. DMA and core accesses to different banks can be completed at full speed, while accesses to the same bank halt the DMA until a program memory slot is available. The Program RAM provides a method of changing the program dynamically, allowing efficient overlaying of DSP software algorithms.

11.1.5.5 Internal Instruction Cache RAM

The program memory space \$000000 – \$00FFFF is for internal Instruction Cache RAM modules (256 locations each). The size of the Instruction Cache is 1024 words (four RAM modules). The starting address of the Instruction Cache space is above the internal Program RAM and is also device-dependent. The Instruction Cache can be disabled by clearing the Cache Enable (CE) bit in the chip Status Register (SR). If the CE bit is cleared, the Instruction Cache RAM becomes the high part of the internal Program RAM. The Instruction Cache is used to minimize contention with accesses to external program memory space. A complete description of the Instruction Cache is provided in **Chapter 8, Instruction Cache**.

11.2 Sixteen-Bit Compatibility Mode

When the Sixteen Bit Compatibility (SC) mode bit is set, the memory map is changed to allow easy access to memory mapped I/O, as described in **Figure 11-2**.

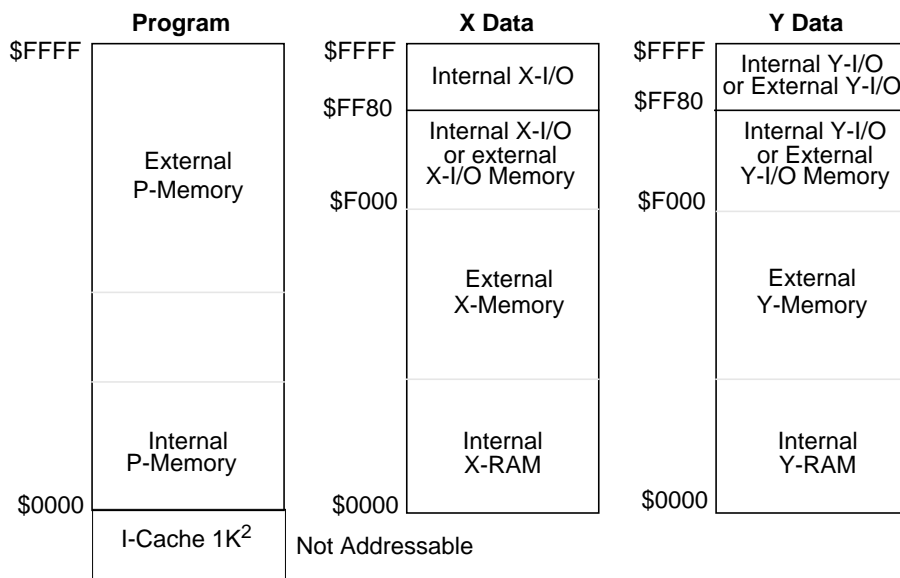


Figure 11-2. DSP56300 Core Memory Map (SC = 1)

For details on this mode, how it affects AGU operations, and functional restrictions, see **Chapter 4, *Address Generation Unit***.

11.3 Memory Switch Mode

When the Memory Switch (MS) mode bit is set, some of the internal data memory addresses (X, Y, or both) become part of the chip internal Program RAM. The addresses are in the higher part of the internal RAM that resides in the lower part of the data memory. The amount of memory transferred is a multiple of 256/1K and is device-dependent.

Due to pipelining, a change in the MS bit takes affect only after the four consecutive instruction cycles. Inserting four NOP instructions after the instruction that changes the value of the MS bit guarantees proper operation.

