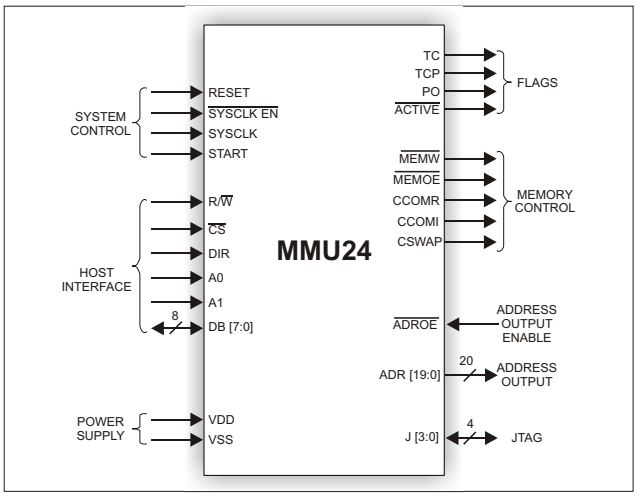
DSP Architectures
Transform Your World™

MMU24

High Performance DSP Memory Management Unit

Data Sheet



FEATURES/BENEFITS:

- ☐ 80/100 MHz operation on complex arrays up to 1 Million words.
- ☐ Complements DSP24 and allows a complex sample rate of 50 MHz for a 1K complex FFT with window
- ☐ Polyphase addressing and control for FFT folding and overlapping
- ☐ Supports DSP, Complex math, Matrix math, FIRs, Block Adds, Subtracts, Complex magnitude, etc.
- ☐ Multiple MMU24s maybe cascaded for increased performance & radices up to Radix 1024
- ☐ User programmable internal loops for virtually any DSP memory address sequence.
- ☐ Supports Radix 2 through Radix 1024 butterfly and twiddle addressing
- ☐ Simple 8-bit host interface reduces host address decoding
- ☐ Support for reduced latency stacked 1-D, 2-D, and 3-D FFT algorithms.
- ☐ Over 240 primitive DSP sequences buried in the chip architecture dramatically reduce software development
- ☐ Advanced 0.5 micron, optional very low power 3.3 volt operation. Extensive built in test including JTAG
- ☐ FFT addressing from any offset address. Enhanced dual (NN) real and double length (2N) real FFT support.
- ☐ ASIC silicon cores, Macro cells, Super cells, and custom constructs available
- ☐ High Reliability and Rad Hard versions available

DESCRIPTION:

The MMU24 Memory Management Unit (MMU) is a high-speed memory addressing device that is uniquely designed to support FFT based approachs to real time DSP applications, see Figure 1.

The MMU24 is a self-contained, small-pin-count device with virtually no computational overhead. The MMU is intended to be used with DSP24 Digital Signal Processor and other compatible execution units. The MMU24 can generate over 240 address patterns. Table 1 summarizes the MMU's address pattern set.

The MMU24 generates a series of addresses for memory arrays. It is easily programmed to generate addressing sequences between four points and one million (1 Meg) points. The MMU24 can be used for standalone operation or as a peripheral on a host CPU bus. The MMU24 contains 32 words of program memory that can be programmed via the 8-bit data bus DB[7:0] (internal memory mode). These 32 words (instructions) of program memory are sufficient to generate algorithms for many applications. For example, a 1 million point, radix-32 FFT can be performed using only four address patterns.

In contrast, in external memory mode, program memory may be bypassed and the MMU24 programmed in external memory mode. For example, an adaptive filtering algorithm that requires thousands of computed passes is one type of algorithm that could be performed in external mode.

Lastly, for unique applications the end user can generate custom address patterns with the MMU24's dual increment sequencer for such things as acquisition data de-interleaving, 2-D and 3-D image corner turning, polyphase overlapping, etc.

Figure 1. FFT Based Approach

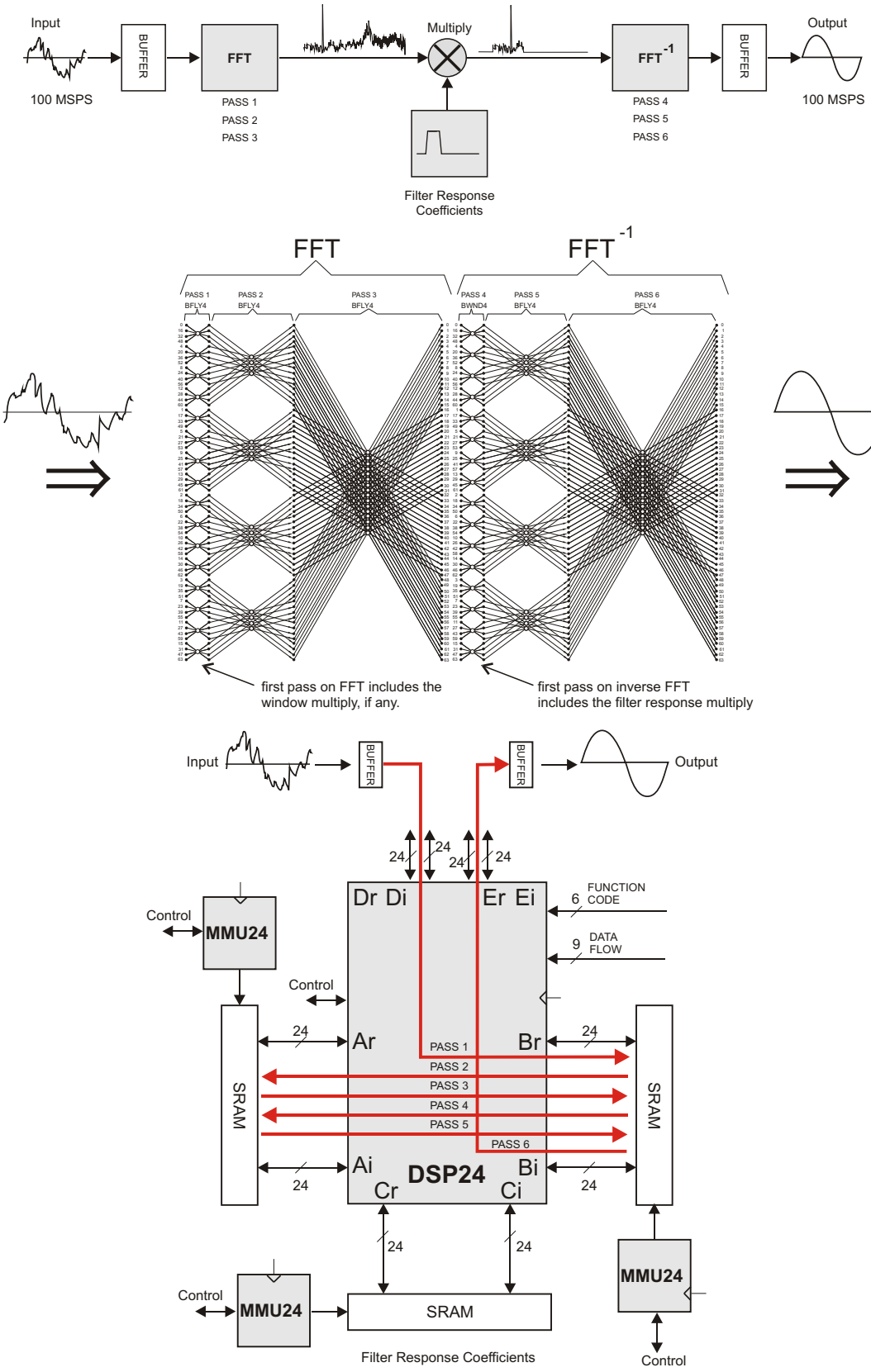


Table 1. MMU24 Address Pattern Set Summary

Opcode		General Purpose/Utility Patterns
00	NOP	No operation
72	INC	Index increment
	MODINC	Replaced with DUALINC patterns
	MODDEC	Replaced with DUALINC patterns
7A	PADHIGH	Pad at the end of a sequence
7B	PADHIGHP	Pad at the end of a sequence using PO flag
98	PADHIGHEP	Pad at the end of a sequence using early PO flag
99	PADHIGHLP	Pad at the end of a sequence using late PO flag
94	PADLOW	Pad at the start of a sequence
95	PADLOWP	Pad at the start of a sequence using PO flag
9A	PADLOWEP	Pad at the start of a sequence using early PO flag
9B	PADLOWLP	Pad at the start of a sequence using late PO flag
	OVERLAP	Replaced with DUALINC patterns
89	DISCARD	Discard
8A	DISCARDP	Discard using PO flag
9C	DISCARDEP	Discard using early PO flag
9D	DISCARDLP	Discard using late PO flag
87	CMAG	Square of magnitude of a complex number
E7	DUALINC1	Dual Increment (Use Configuration Register Set #1)
E8	DUALINC1P	Dual Increment with PO enabled. (Set SKEW PO High/Low for polarity)
E9	DUALINC1B	Dual Increment with BitReverse enabled
EA	DUALINC1Q	Dual Increment with Quadrant (DEG90) logic enabled
EB	DUALINC1QNS	Dual Increment with Quadrant logic enabled (Turn off SWAP). Use for windowing.
EC	DUALINC1BQ	Dual Increment with Quadrant and BitReverse enabled. Use for windowing
ED	DUALINC1BQNS	Dual Increment with Quadrant, BitReverse(Turn off SWAP). Use for windowing.
EE	DUALINC2	Dual Increment (Use Configuration Register Set #2)
EF	DUALINC2P	Dual Increment with PO enabled. (Set SKEW PO High/Low for polarity)
F0	DUALINC2B	Dual Increment with BitReverse enabled
F1	DUALINC2Q	Dual Increment with Quadrant (DEG90) logic enabled
F2	DUALINC2QNS	Dual Increment with Quadrant logic enabled (Turn off SWAP). Use for windowing.
F3	DUALINC2BQ	Dual Increment with Quadrant and BitReverse enabled. Use for windowing.
F4	DUALINC2BQNS	Dual Increment with Quadrant, BitReverse (Turn off SWAP). Use for windowing.
FE	CLEARSIG	Clear Signature register
FF	VIEWSIG	View Signature register

Table 1. MMU24 Address Pattern Set Summary (cont.)

Opcode		Digit Reverse Patterns
71	RBF0	Digit reversed data address column 0

Opcode		Compatible Patterns
38 to 70	TFx,MXBx,MXTx	Compatible Patterns
15 to 23	BFx	Compatible Patterns

Opcode		Recombine Patterns
8B	BRFTL	2 at-a-time real FFT separation pass (2N point load)
8C	BRFTLS	2 at-a-time real FFT separation pass (N point load)
8D	BRFTU	2 at-a-time real FFT separation pass (2N point unload)
8E	BRFTUS	2 at-a-time real FFT separation pass (2N point unload, Store in N)
8F	BFCTL	Fast cosine transform separation pass data addresses (2N point load)
90	BFCTT	Fast cosine transform separation pass twiddle addresses (2N points)
91	BFCTUS	Fast cosine transform separation pass data addresses (2N point unload, Store in N)
92	BFCTU	Fast cosine transform separation pass data addresses (2N point unload, Store in 2N)
E6	BFCTUS2	Fast cosine transform separation pass data addresses (N point unload, Store in N)
93	BFCTUP	Fast cosine transform separation pass data addresses (2N point unload using PO flag)
9E	BFCTUEP	Fast cosine transform separation pass data addresses (2N point unload using early PO)
9F	BFCTULP	Fast cosine transform separation pass data addresses (2N point unload using late PO)

Table 1. MMU24 Address Pattern Set Summary (cont.)

Opcodes		DSP Patterns
01 to 14	BFCn (0 to 19)	Input/Output Butterfly Address Column n (0 TO 19)
24 to 37	TF2Cn (0 to 19)	Radix 2 twiddle factor Address Column n (0 to 19)
A0 to B2	TF4Cn (0 to 18)	Radix 4 twiddle factor Address Column n (0 to 18)
B3 to C4	TF8Cn (0 to 17)	Radix 8 twiddle factor Address Column n (0 to 17)
C5 to D5	TF16Cn (0 to 16)	Radix 16 twiddle factor Address Column n (0 to 16)
D6 to E5	TF32Cn (0 to 15)	Radix 32 twiddle factor Address Column n (0 to 15)

Opcodes		DSP Patterns (cont)
24 to 37	TF64Cn (0 to 19)	Radix 64 twiddle factor Address Column n (0 to 19)
A0 to B2	TF128Cn (0 to 18)	Radix 128 twiddle factor Address Column n (0 to 18)
B3 to C4	TF256Cn (0 to 17)	Radix 256 twiddle factor Address Column n (0 to 17)
C5 to D5	TF512Cn (0 to 16)	Radix 512 twiddle factor Address Column n (0 to 16)
D6 to E5	TF1024Cn (0 to 15)	Radix 1024 twiddle factor Address Column n (0 to 15)

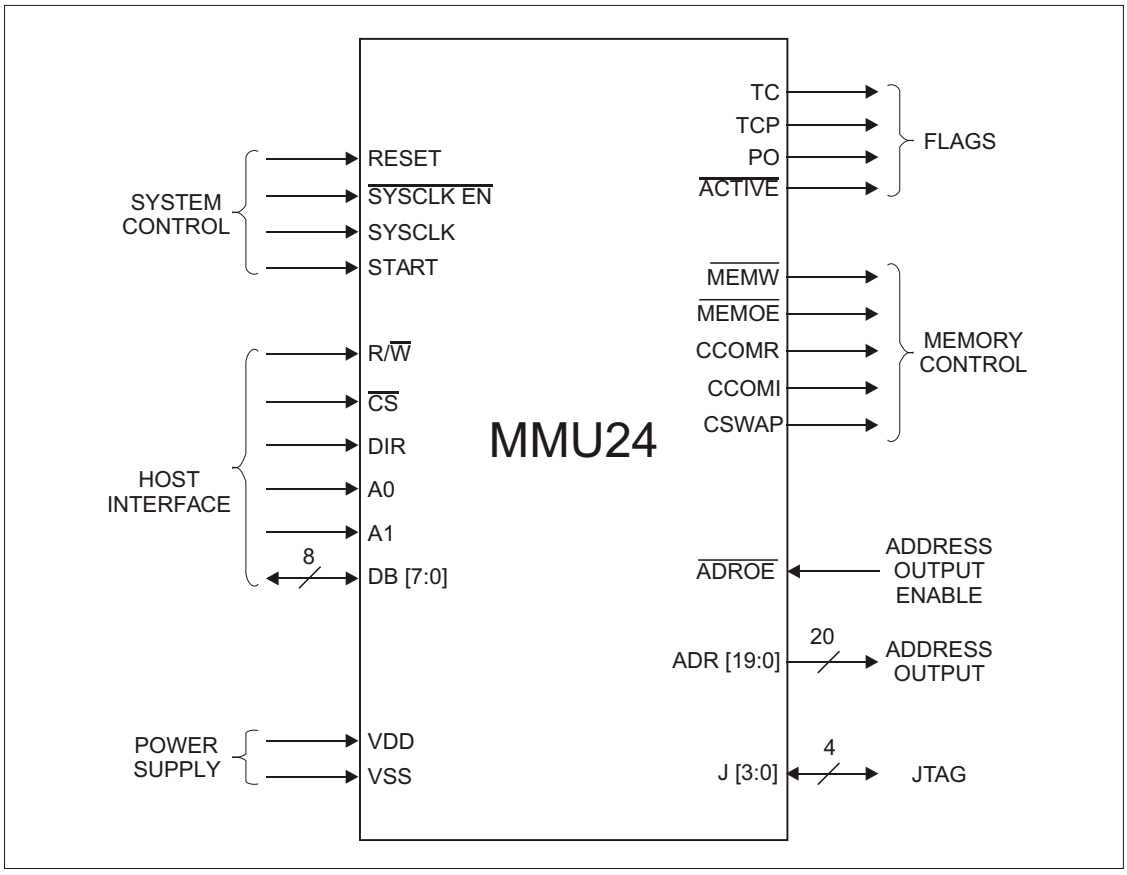


Figure 2. MMU24 Interface Signals

Table 2. MMU24 Interface Signals

SIGNAL	DIRECTION	SIGNAL NAME
DB[7:0]	I/O	Host Interface Data Bus
A0,A1	I	Host Control
R/$\overline{W}$	I	Read/Write
$\overline{CS}$	I	Chip Select
DIR	I	Host Interface Direction
RESET	I	Chip Reset
SYSCLK	I	System Clock
START	I	Start of Pass
$\overline{SYSCLKEN}$	I	Pause Address Generation when inactive (HIGH)
TC	O	Terminal Count
TCP	O	Programmable TC
ADR[19:0]	O	Address Outputs
$\overline{ADROE}$	I	Address Output Enable
CCOMR	O	Coefficient Complement Real (data to DSP24)
CCOMI	O	Coefficient Complement Imaginary (data to DSP24)
SWAP	O	Swap Real and Imaginary (data to DSP24)
PO	O	Programmable Output
$\overline{MEMW}$	O	Memory Write
$\overline{MEMOE}$	O	Memory Output Enable
$\overline{ACTIVE}$	O	User Programmable Flag
VDD	P	+3.3 Volt Power Supply
VSS	P	Chip Ground
JDO	O	JTAG Test Signal
JDI	I	JTAG Test Signal
JMS	I	JTAG Test Signal
JCK	I	JTAG Test Signal
$\overline{JRST}$	I	JTAG Test Signal

INTERFACE SIGNALS

The MMU24 interface signals are shown in Figure 2 and summarized in Table 2.

DB[7:0] (Host Interface Data Bus)

In internal memory mode, the 8-bit data bus, DB[7:0], is used to load MMU24 registers via a two-step scheme: First write the address of the target register, then write the data. In external memory mode, the register address is set to one of three predefined locations based on the A1, A0 signals. In this case, the DB[7:0] data bus transfers data to the selected predefined location.

A1,A0 (Host Control)

For internal memory mode, the A1,A0 signals define the input to the DB[7:0] data bus as:

Address to a target register (A1 is LOW, A0 is HIGH)

Data to a target data (A1 is LOW, A0 is LOW).

In external memory mode, A1,A0 sets the register address to one of the following predefined address location:

Location 31 of program memory (Host Register Address 1F)

Location 31 of pipeline/latency memory (Host Register Address 3F)

MODE register (Host Register Address A2)

Refer to Host Interface Table 2 for more information.

R/$\overline{W}$ (Read/Write Host)

R/$\overline{W}$ loads each Host register on its rising edge, effectively becoming the clock for all Host registers.

Note: R/$\overline{W}$ loads the Host registers. The SYSCLK signal controls the execution and output of the MMU24 after the Host is loaded. The R/$\overline{W}$ and SYSCLK signals do NOT have to be synchronous with one another. However, the START signal must be synchronized to the SYSCLK.

$\overline{CS}$ (Chip Select)

When set LOW, $\overline{CS}$ enables the MMU24 to communicate with the Host to program the registers. When $\overline{CS}$ is set HIGH, DB[7:0] is disabled to a high-impedance state regardless of the activity of the R/$\overline{W}$ signal.

DIR (Read/Write Direction of the DB[7:0] Host Interface Data Bus)

When DIR is set LOW, the DB[7:0] bus is tri-stated, and writing of address and data values are allowed.

When DIR is set HIGH, the DB[7:0] bus is driven by the MMU24 with the register value located at the last written address location.

RESET (Chip Reset)

When set HIGH, RESET does a hardware reset of the MMU24 chip. When set LOW, normal operation of the MMU24 chip is enabled.

SYSCLK (System Clock)

SYSCLK controls the execution and output of the address calculation logic. A new address is generated on each rising edge of the SYSCLK signal. Refer to Address Calculation for more information.

Note: The START signal must be synchronized to SYSCLK.

START (Start of a Pass)

START is an edge-triggered signal that initiates the generation of each addressing sequence. After the initial START, the MMU24 requires seven SYSCLK cycles of latency before the first valid address is output. Each subsequent START can be set to zero latency provided START is issued before the end of the current sequence.

Note: START must be synchronized to the SYSCLK signal and must not be issued at least two SYSCLK cycles after the last write to the host memory.

START for "Pre-Starting" must go active at least 6 cycles before last address.

START for "Pre-Starting" must NOT go active on the cycle following TC rising edge.

External programming must be finished at least 7 cycles before end of current pattern.

$\overline{SYSCLKEN}$

An active HIGH on the $\overline{SYSCLKEN}$ signal causes the current address generation to stop, and continuously output the last value. When $\overline{SYSCLKEN}$ returns inactive (LOW), the address generation will resume.

TC (Terminal Count)

TC indicates when an addressing sequence has been completed. By default, TC goes HIGH one cycle before the end of the current addressing sequence. TC remains HIGH until the next START signal causes a valid address to be generated. The LOW to HIGH transition of TC can be delayed by one cycle (to the actual end of the sequence) by setting bit 3 in the MODE register to HIGH. Refer to MODE Register. Once the transitions of TC are defined, they may be skewed together by a value in the SKEW register.

The TC region of activity may be skewed by -1 to +6 SYSCLK cycles from its normal offset of 0 as defined by the SKEW register.

Note: The $\overline{\text{PAUSE}}$ register affects The TC signal. If PAUSE is used, TC transitions low to high one cycle before the sequence length + $\overline{\text{PAUSE}}$.

TCP (Programmable TC)

This signal is equivalent to the TC signal, with the additional capability of allowing the user to invert the output polarity, or to cause it to act like a mini-TC for address sequences that are executed back-to-back via the MASTERREPEAT register. (Strobe LOW and then HIGH one address before the last address out for each of the individual address sequences.)

The controls for delaying TCP by a SYSCLK cycle, for inversion, and for strobing during a MASTERREPEAT are located in the SKEW register.

The TCP region of activity may be skewed by -1 to +6 SYSCLK cycles from its normal offset of 0 as defined by the SKEW register.

Note: TCP is set HIGH after a reset.

ADR[19:0] Address Outputs

ADR[19:0] are the address outputs from the MMU24. They may be Tri-Stated with the ADDROE signal.

$\overline{\text{ADROE}}$ (Address Output Enable)

This signal controls how the ADR[19:0] are driven. When active LOW, the ADR[19:0] signals are driven. When inactive HIGH, the ADR[19:0] signals are Tri-Stated.

CCOMR, CCOMI, SWAP (Coefficient Complement Real and Imaginary, and Swap Real/Imaginary)

These three signals can be used to manipulate coefficient data when they are connected to the DSP24 corresponding inputs. They indicate functions that should be applied to the data input to the DSP24 coefficient port in order to facilitate storage of Twiddle Coefficient data for FFT's in 1/4 of the normal required storage space or Window data in 1/2 of the normal required storage space. These signals are activated only when bit 4 in the MODE register is set HIGH and a twiddle address pattern is used. All signals are Tri-Stated when not used.

CCOMR, CCOMI, and SWAP are generated from the quadrant in the lookup table that the output address is pointing to. The MEMSIZE register defines the size of the lookup table.

An Active HIGH level on the coefficient complement real (CCOMR) signal indicates that the real coefficient point should be complemented.

An Active HIGH level on the coefficient complement imaginary (CCOMI) signal indicates that the real coefficient point should be complemented.

An Active HIGH level on the SWAP signal indicates that the real and imaginary Coefficient data should be exchanged. If active at the same time as CCOMR or CCOMI, the CCOMR/CCOMI operation should be performed first (This is the order of precedence the DSP24 uses.)

The CCOMR, CCOMI, and SWAP signals region of activity may be skewed by 0 to +6 SYSCLK cycles from its normal offset of 0 as defined by the SKEW register.

PO (Programmable Output)

PO indicates when an index address is being produced. When HIGH, the PO signal indicates that the actions shown in Table 3 need to be performed.

Table 3. PO Signal Selections

PADHIGHP	Data needs to be padded
PADHIGHEP	Data for the next address needs to be padded
PADHIGHLP	Data for the previous address neeeds to be padded
PADLOWP	Data needs to be padded
PADLOWEP	Data for the next address needs to be padded
PADLOWLP	Data for the previous address needs to be padded
DISCARDP	Data needs to be discarded
DISCARDEP	Data for the next address needs to be discarded
DISCARDLP	Data for the previous address needs to be discarded
BFCTUP	Data for the highest frequency component needs to be discarded
BFCTUEP	Data for the next address (highest frequency component) needs to be written
BFCTULP	Data for the previous address (highest frequency component) needs to be written
DUALINC1P	User defined flag for custom solutions
DUALINC2P	User defined flag for custom solutions

The PO region of activity may be skewed by -1 to +6 SYSCLK cycles from its normal offset of 0 as defined by the SKEW register.

The phase of PO (PADHIGH or PADLOW) is defined by the SKEW register for the DUALINC1P and DUALINC2P address patterns.

Note: after a reset or if PO is not used by a valid addressing sequence, this signal is set to high-impedance.

$\overline{\text{MEMW}}$ (Memory write)

$\overline{\text{MEMW}}$ indicates when the execution unit is to write to the SRAM. This signal is controlled by bit 7 of the pipeline/memory latency memory. $\overline{\text{MEMW}}$ is pulsed LOW when data is to be written to the SRAM. $\overline{\text{MEMW}}$ is set HIGH when data is read from the SRAM

The $\overline{\text{MEMW}}$ region of activity may be skewed by -1 to +6 SYSCLK cycles from its normal offset of 0 as defined by the SKEW register.

Note: $\overline{\text{MEMW}}$ is set inactive HIGH after a reset.

$\overline{\text{MEMOE}}$ (Memory Output Enable)

$\overline{\text{MEMOE}}$ controls the output enable of the SRAM. This signal is controlled by bit 7 of the pipeline/memory latency memory. $\overline{\text{MEMOE}}$ is set LOW when data is to be read from the SRAM. $\overline{\text{MEMOE}}$ is set HIGH when data is to be written to the SRAM.

The $\overline{\text{MEMOE}}$ region of activity may be skewed by -1 to +6 SYSCLK cycles from its normal offset of 0 as defined by the SKEW register.

Note: $\overline{\text{MEMOE}}$ is set inactive HIGH after a reset.

$\overline{\text{ACTIVE}}$

$\overline{\text{ACTIVE}}$ is a user-defined flag that can be defined on a pass basis alongside the Function Code and Latency. When enabled, the $\overline{\text{ACTIVE}}$ signal will go LOW with the first address output, and HIGH the cycle after the last address, effectively framing the addresses of interest.

The $\overline{\text{ACTIVE}}$ region of activity may be skewed by -1 to +6 SYSCLK cycles from its normal offset of 0 as defined by the current instruction pointer to the extended feature memory.

Note: $\overline{\text{ACTIVE}}$ is set inactive HIGH after a reset.

VDD

VDD is the power supply for the chip

VSS

VSS is the ground for the chip.

J[3:0]

JTAG chip test pins.

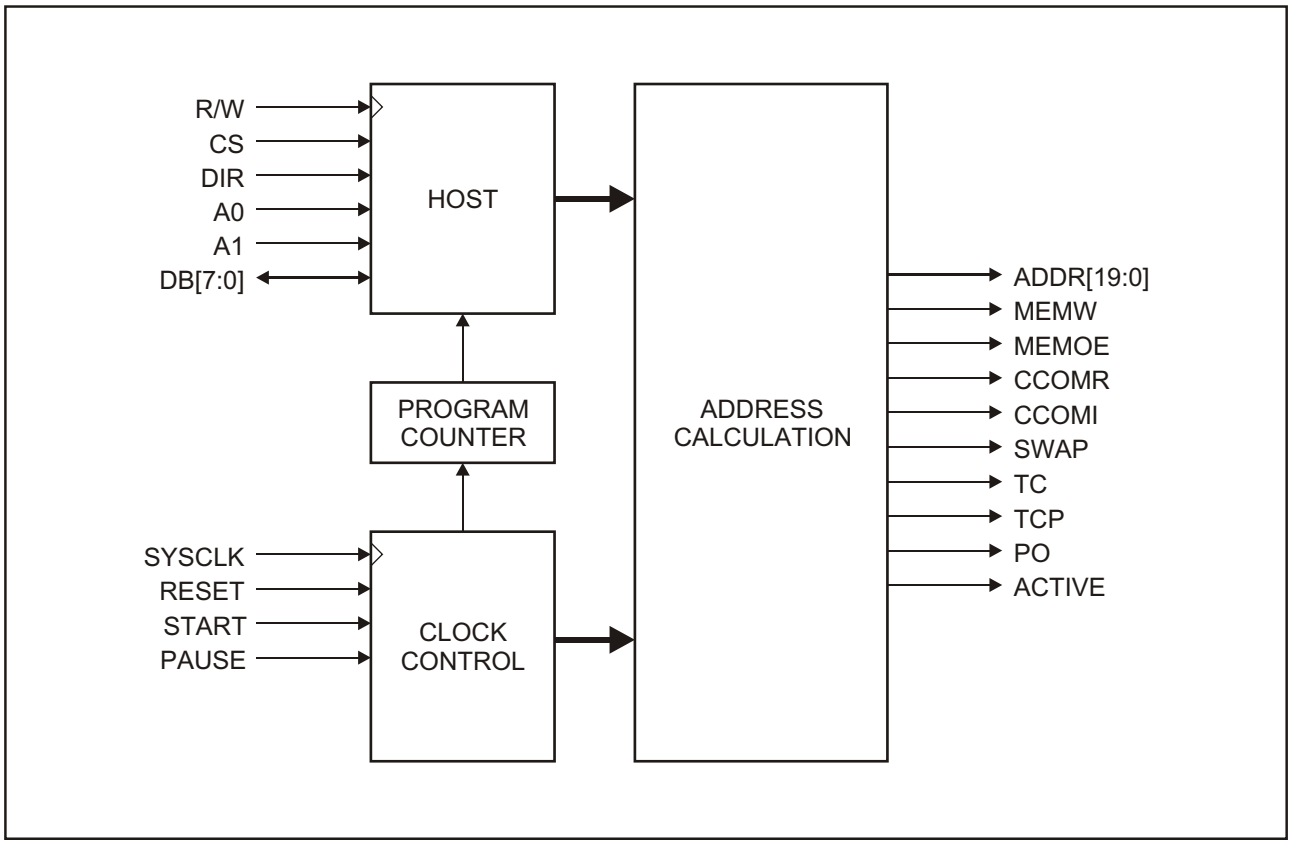


Figure 3. MMU24 Simplified Block Diagram

BLOCK LEVEL DESCRIPTION

MMU24 Block Diagram

The MMU24 Memory Manager consists of the following functional sections: Host, Address Calculation, Clock Control, and Program Counter. See Figure 3.

HOST

The Host Interface allows the MMU24 to be programmed and read from a simple 8-bit data bus. Through this data bus, the user is able to access any of the 214 internal memory locations.

The host memory is segmented into two major groups of registers, which are the execution registers, and the configuration registers.

See Host Memory Map Figure 4.

The execution registers are made up of the Program Memory, Latency Memory, and the Extended Feature Memory register files. Each of the locations of the register files may be programmed or read in any order, however, when the MMU24 is started (via START) only one value from each of these registers is used at a time. The values that are used are determined by the current Program Counter's index.

The configuration registers make up the remainder of the registers. These registers differ from the execution registers in that multiple registers may be needed at any one time depending on the address pattern being executed. See Figure 7.

REGISTER DEFINITIONS

Program Memory

The program memory contains 32, 8-bit registers (locations 0-1F) that can store up to 32 address patterns.

Pipeline/Memory Latency Memory

The pipeline/memory latency memory contains 32, 8-bit registers (locations 20-3F) that store the latency value for each address pattern. The latency value defines when the related address pattern is to be generated. Memory latency values can range from 0 to 127 (bits[6:0]). However, by setting bit 5 of the mode register to HIGH, the latency numbers can be multiplied by 2, thus extending the latency range to 254. Bit 7 of the pipeline/memory latency memory indicates that the MMU24 is reading from or writing to the SRAM. Refer to MEMOE (Memory Output Enable) Signal, MEMW (Memory Write) Signal, and to MODE Register. See LATENCY Register Figure 5.

Extended Feature Memory

The Extended Feature memory is divided into two banks of 32. The first bank contains 32, 8-bit registers (locations 40-5F) that store 5 additional MSB bits for the latency value for each address pattern. The upper 3 MSB's define the relative skew (-1 to +6 SYSCLK cycles from its normal offset of 0) for the $\overline{\text{ACTIVE}}$ signal.

The second bank contains 32, 1-bit registers (locations 60-7F) that store the enable control for the ACTIVE signal. See EXTENDED MEMORY Register Figure 5.

60-7F) that store the enable control for the $\overline{\text{ACTIVE}}$ signal. See EXTENDED MEMORY Register Figure 5.

Configuration Registers

The configuration registers (A0-F6) comprise the remainder of the MMU24 Host Memory

PCSTART Register

The 5-bit PCSTART register (A0) points to the first address pattern to be executed in program memory or where the next address pattern will come from if the current address pattern equals PCEND.

In external memory mode, PCSTART is disabled and the first address pattern to be executed points to location 31 of the program memory.

PCEND Register

The 5-bit PCEND register (D3) points to the last address pattern to be executed in program memory. In external memory mode, PCEND is disabled and the last address pattern to be executed points to location 31 of program memory.

MODE Register

The 8-bit Mode register (A2) defines the various modes of operation of the MMU24. The bits for the MODE register shown in Figure 4 are defined as follows:

Bit 0, Determines when the MMU24 is in internal or external memory mode (internal is LOW, external is HIGH).

Bit 1, Enables the BREAKPOINT register (enable is HIGH, disable is LOW).

Bit 2, Enables the DIGITREV logic. (enable is HIGH, normal operation is LOW).

Bit 3, Delays the Low-to-High transition of TC by 1 cycle to the actual end of the sequence (1 cycle delay is HIGH, no delay is LOW). If delayed, TC goes HIGH at the same time that the last address is generated. Bit 3 does not affect the falling edge of TC.

Bit 4, Enables the Quadrant logic to force the output address into 90 degrees of the MEMSIZE sized lookup table and signal which quadrant the address is in via the CCOMR, CCOMI, and SWAP signals.

Bit 5, Indicates when the values in the Latency Memory[12:0] are multiplied by 2 (Multiplied by 2 is HIGH).

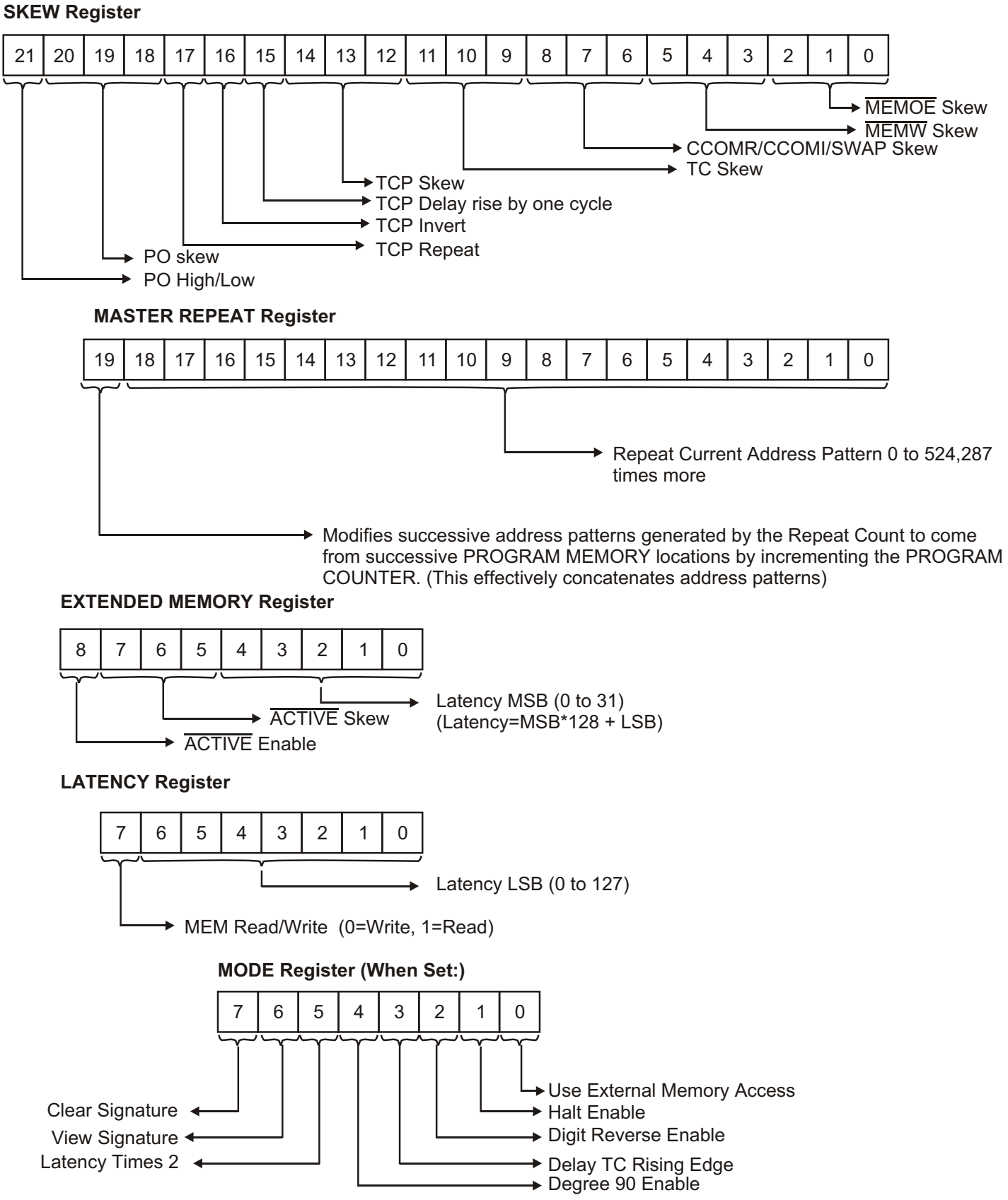
Bit 6, Enables the output of the signature register on the ADDR[19:0] lines when active HIGH.

Bit 7, Clears the signature register when active HIGH.

Figure 4. Host Memory Map

Program Memory	0	ADDRESS PATTERN 0	AF	ADR2START [7:0]	D3	PCEND [4:0]
	1	ADDRESS PATTERN 1	B0	ADR2START [15:8]	D4	HALT [0]
		:	B1	ADR2START [19:16]	D5	DIGITREV [7:0]
	1F	ADDRESS PATTERN 31	B2	DFACTOR / ADR2LENGTH [7:0]	D6	DIGITREV [15:8]
Latency Memory	20	LATENCY FOR PATTERN 0	B3	DFACTOR / ADR2LENGTH [15:8]	D7	DIGITREV [19:16]
	21	LATENCY FOR PATTERN 1	B4	DFACTOR / ADR2LENGTH [19:16]	D8	MEMSIZE [7:0]
		:	B5	LEAP / ADR2INC [7:0]	D9	MEMSIZE [15:8]
	3F	LATENCY FOR PATTERN 31	B6	LEAP / ADR2INC [15:8]	DA	MEMSIZE [20:16]
Extended Feature Memory	40	EXT FEAT [7:0] FOR PATTERN 0	B7	LEAP / ADR2INC [19:16]	DB	CHIPID [7:0]
	41	EXT FEAT [7:0] FOR PATTERN 1	B8	NLEAP / INDEX2LENGTH [7:0]	DC	DISCARD / DINC1PAIRS [7:0]
		:	B9	NLEAP / INDEX2LENGTH [15:8]	DD	DISCARD / DINC1PAIRS [15:8]
	5F	EXT FEAT [7:0] FOR PATTERN 31	BA	NLEAP / INDEX2LENGTH [19:16]	DE	DISCARD / DINC1PAIRS [19:16]
	60	EXT FEAT [8:8] FOR PATTERN 0	BB	INDEX2START [7:0]	DF	INCADRSTART [7:0]
	61	EXT FEAT [8:8] FOR PATTERN 1	BC	INDEX2START [15:8]	E0	INCADRSTART [15:8]
		:	BD	INDEX2START [19:16]	E1	INCADRSTART [19:16]
	7F	EXT FEAT [8:8] FOR PATTERN 31	BE	ZEROPAD / INDEX2INC [7:0]	E2	INCINDEXSTART [7:0]
	80	NOT USED	BF	ZEROPAD / INDEX2INC [15:8]	E3	INCINDEXSTART [15:8]
	81	NOT USED	C0	ZEROPAD / INDEX2INC [19:16]	E4	INCINDEXSTART [19:16]
		NOT USED	C1	ADRSTART [7:0]	E5	INCADR2START [7:0]
		:	C2	ADRSTART [15:8]	E6	INCADR2START [15:8]
	9F	NOT USED	C3	ADRSTART [19:16]	E7	INCADR2START [19:16]
	A0	PCSTART [4:0]	C4	ADRLENGTH [7:0]	E8	INCINDEX2START [7:0]
	A1	NOT USED	C5	ADRLENGTH [15:8]	E9	INCINDEX2START [15:8]
	A2	MODE [7:0]	C6	ADRLENGTH [19:16]	EA	INCINDEX2START [19:16]
	A3	OVERLAP / DINC2PAIRS [7:0]	C7	ADRINC [7:0]	EB	MASTER OFFSET [7:0]
	A4	OVERLAP / DINC2PAIRS [15:8]	C8	ADRINC [15:8]	EC	MASTER OFFSET [15:8]
	A5	OVERLAP / DINC2PAIRS [19:16]	C9	ADRINC [19:16]	ED	MASTER OFFSET [19:16]
	A6	PAUSE [7:0]	CA	ADRDEC / INDEXLENGTH [7:0]	EE	MASTEROFFINC [7:0]
	A7	PAUSE [15:8]	CB	ADRDEC / INDEXLENGTH [15:8]	EF	MASTEROFFINC [15:8]
	A8	PAUSE [19:16]	CC	ADRDEC / INDEXLENGTH [19:16]	F0	MASTEROFFINC [19:16]
	A9	BREAKPOINT [7:0]	CD	INDEXSTART [7:0]	F1	MASTERREPEAT [7:0]
	AA	BREAKPOINT [15:8]	CE	INDEXSTART [15:8]	F2	MASTERREPEAT [15:8]
	AB	BREAKPOINT [19:16]	CF	INDEXSTART [19:16]	F3	MASTERREPEAT [19:16]
	AC	N [7:0]	D0	INDEXINC [7:0]	F4	SKEW [7:0]
	AD	N [15:8]	D1	INDEXINC [15:8]	F5	SKEW [15:8]
	AE	N [20:16]	D2	INDEXINC [19:16]	F6	SKEW [21:16]

Figure 5. Register Bit Definitions



DINC2PAIRS Register

The 20-bit (two 8-bit and one 4-bit register) register (A3-A5) is used by the DUALINC2/P/B/Q/QNS/BQ/BQNS address patterns.

When used by the DUALINC2/P/B/Q/QNS/BQ/BQNS address patterns, this register defines the number of times to cycle the addressing of each pair of sub-increment (ADRx and INDEXx) address patterns.

DISCARD / DINC1PAIRS Register

The 20-bit (two 8-bit and one 4-bit register) register (DC-DE) is used by the DISCARD/P/EP/LP and DUALINC1/P/B/Q/QNS/BQ/BQNS address patterns.

When used by the DISCARD/P/EP/LP address patterns, this register indicates the number of addresses to be discarded (00000 H to FFFFF H). (If the P/EP/LP versions are used, PO will signal this by going HIGH)

When used by the DUALINC1/P/B/Q/QNS/BQ/BQNS address pattern, this register defines the number of times to cycle the addressing of each pair of sub-increment (ADRx and INDEXx) address patterns.

PAUSE Register

The 20-bit (two 8-bit and one 4-bit register) PAUSE register (A6-A8) defines the number of cycles the MMU24 is to pause immediately after an addressing sequence is completed. If used, the effective length (N, ADRLENGTH or 2N) is changed to the sequence length + PAUSE. In contrast to the LATENCY MEMORY, which has a unique value for each address pattern, PAUSE has a global value for all address patterns and only extends the end of the address pattern not the start.

The PAUSE register affects the TC signal. If PAUSE is used, TC becomes active one cycle before sequence length + PAUSE.

Note: if MODE register bit 3 is set HIGH, TC transitions at the same time as sequence length + PAUSE.

BREAKPOINT Register

The 20-bit (two 8-bit and one 4-bit register) BREAKPOINT register (A9-AB) defines the break point for an addressing sequence (halts after a specific number of addresses are output). The BREAKPOINT register is used only when bit 1 in the MODE register is set HIGH.

If used, the value in the BREAKPOINT register becomes the new length of the current sequence (instead of N, ADRLENGTH, or 2N). When the number of addresses generated in the sequence reaches the value in the BREAKPOINT register, the addressing sequence ends. In turn, the TC signal becomes active one cycle before the value in the BREAKPOINT register is reached. The BREAKPOINT stops the current addressing sequence provided that the sequence has not already ended and has not entered a PAUSE state.

one cycle before the value in the BREAKPOINT register is reached. The BREAKPOINT stops the current addressing sequence provided that the sequence has not already ended and $\overline{\text{SYSCLKEN}}$ is low.

N Register

The 21-bit (two 8-bit and one 5-bit register) N register (AC-AE) defines the length of the addressing sequence for most address patterns. In contrast, some address patterns use a sequence length of 2N, such as BRFTL and BRFTU, or use a value in the ADRLENGTH resister instead of N, such as MODINC and MODDEC, or use a combination of (ADRLENGTH+INDEXLENGTH)*DINCPAIRS, such as DUALINC1x and DUALINC2x .

Note: because internally the MMU24 requires the use of all values of N which are divisible by powers of 2 (N/2, N/4, N / 8 , …), the N register is 21 bits wide. The multiples of N are generated internally using a shift right. N must always be a power of 2 between 000004 H and 100000 H

ADR2START Register

The 20-bit (two 8-bit and one 4-bit register) ADR2START register (AF-B1) is used by the DUALINC2/P/B/Q/QNS/BQ/BQNS address patterns. The ADR2START register defines the starting address for the first of two increment sequences. See Figure 6.

DFACTOR (Decimate Factor) / ADR2LENGTH Register

The 20-bit (two 8-bit and one 4-bit register) register (B2-B4) is used by the DECIM and ADECIM address patterns, and the DUALINC2/P/B/Q/QNS/BQ/BQNS address patterns:

When used by DECIM address pattern, the addressing sequence will start at address 0, and will continue accumulating by the amount given by the DFACTOR. The sequence will stop when the number of addresses generated reaches N.

When used by ADECIM address pattern, the DFACTOR value is used as in the DECIM pattern, but only for the first N addresses. Then, DFACTOR will be internally accumulated (to DFACTOR = DFACTOR + LEAP) for the next N addresses, etc.

When used by the DUALINC2/P/B/Q/QNS/BQ/BQNS address patterns, the ADR2LENGTH register defines the number of addresses to output for the first of two increment sequences.

LEAP / ADR2INC Register

The 20-bit (two 8-bit and one 4-bit register) register (B5-B7) is used by the ADECIM and DUALINC2/P/B/Q/QNS/BQ/BQNS address patterns.

When used by the ADECIM address pattern, LEAP defines the accumulation amount for the DFACTOR value. The DFACTOR value is increased by LEAP for each set of N addresses.

When used by the DUALINC2/P/B/Q/QNS/BQ/BQNS address patterns, the ADR2INC register defines the amount to increment each subsequent address by for the first of two increment sequences.

NLEAP / INDEX2LENGTH Register

The 20-bit (two 8-bit and one 4-bit register) register (B8-BA) is used by the ADECIM and DUALINC2/P/B/Q/QNS/BQ/BQNS address patterns.

When used by the ADECIM address pattern, NLEAP defines the number of sets of N addresses that are to be generated.

When used by the DUALINC2/P/B/Q/QNS/BQ/BQNS address patterns, the INDEX2LENGTH register defines the number of addresses to output for the second of two increment sequences.

INDEX2START Register

The 20-bit (two 8-bit and one 4-bit) register (BB-BD) is used by the DUALINC2/P/B/Q/QNS/BQ/BQNS address patterns to define the starting address for the second of two increment sequences.

ZEROPAD / INDEX2INC Register

The 20-bit (two 8-bit and one 4-bit register) register (BE-C0) is used by the PADLOW/P/EP/LP, PADHIGH/P/EP/LP, and DUALINC2/P/B/Q/QNS/BQ/BQNS address patterns.

When used by the PADLOW/P/EP/LP address patterns, the ZEROPAD register indicates the number of INDEX addresses that are to be generated before the actual sequence starts.

When used by the PADHIGH/P/EP/LP address patterns, the ZEROPAD register indicates when the INDEX address is to be generated to replace the current sequence.

Note: The INDEX address starts at INDEXSTART and increments by INDEXINC. Also, when ZEROPAD = 0 or ZEROPAD = ADRLENGTH, instead of having a 0 length/non-existent index/address pattern, the routine will interpret the 0 length as equal to 1 million.

When used by the DUALINC2/P/B/Q/QNS/BQ/BQNS address patterns, the INDEX2INC register defines the amount to increment each subsequent address by for the second of two increment sequences.

ADRSTART Register

The 20-bit (two 8-bit and one 4-bit register) ADRSTART register (C1-C3) is used by PADLOW/P/EP/LP, PADHIGH/P/EP/LP, MODINC, MODDEC, INC, and DUALINC1/P/B/Q/QNS/BQ/BQNS address patterns to define the starting address in the sequence. (or the starting address of the first of two address sequences for the DUALINC1/P/B/Q/QNS/BQ/BQNS address patterns.)

ADRLENGTH Register

The 20-bit (two 8-bit and one 4-bit register) ADRLENGTH register (C4-C6) is used by PADLOW/P/EP/LP, PADHIGH/P/EP/LP, MODINC, MODDEC, INC, and DUALINC1/P/B/Q/QNS/BQ/BQNS address patterns to indicate the number of points the sequence contains. Unlike the N register, ADRLENGTH does not have to be a power of 2. (ADRLENGTH defines the starting address of the first of two address sequences for the DUALINC1/P/B/Q/QNS/BQ/BQNS address patterns.

Note: if a length of 1 million (1 mega word) is required, the ADRLENGTH register must be set to zero.

ADRINC (Address Increment) Register

The 20-bit (two 8-bit and one 4-bit register) ADRINC register (C7-C9) is used by PADLOW/P/EP/LP, PADHIGH/P/EP/LP, MODINC, INC, and DUALINC1/P/B/Q/QNS/BQ/BQNS address patterns to indicate the amount by which each subsequent address is to be incremented. (For the fist of two address sequences for the DUALINC1/P/B/Q/QNS/BQ/BQNS address patterns.

ADRDEC (Address Decrement) / INDEXLENGTH Register

The 20-bit (two 8-bit and one 4-bit register) register (CA-CC) is used by the MODDEC and DUALINC1/P/B/Q/QNS/BQ/BQNS address patterns.

When used by the MODDEC address pattern, the ADRDEC register indicates the amount by which each address is to be decremented.

When used by the DUALINC1/P/B/Q/QNS/BQ/BQNS address patterns, the INDEXLENGTH register defines the number of addresses to output for the second of two increment sequences.

INDEXSTART Register

The 20-bit (two 8-bit and one 4-bit register) INDEXSTART register (CD-CF) is used by the PADLOW/P/EP/LP, PADHIGH/P/EP/LP, DISCARD/P/EP/LP, BFCTU/P/EP/LP, and DUALINC1/P/B/Q/QNS/BQ/BQNS address patterns.

While the exact application for each address pattern differs, the INDEXSTART value (00000 H to FFFFF H) can point to an address space outside of the working space memory for a particular address pattern.

Note: *if the address patterns that use the indexing address are used on the input side of the execution unit, data must be written to the INDEXSTART location prior to running the address pattern. If INDEXSTART is used on the output side of the execution unit, the PO signal can be used in conjunction with the indexing address to determine when the data is to be written to the RAM.*

When used the by the DUALINC1/P/B/Q/QNS/BQ/BQNS address patterns, INDEXSTART defines the starting address for the second of two increment sequences. address for the second of two increment sequences.

INDEXINC Register

The 20-bit (two 8-bit and one 4-bit register) INDEXINC register (D0-D2) is used by the PADLOW/P/EP/LP, PADHIGH/P/EP/LP, and DUALINC1/P/B/Q/QNS/BQ/BQNS address patterns. The INDEXINC value (00000 H to FFFFF H) identifies the amount by which the INDEXADR value is to be incremented each time an indexing address is generated.

When used by the DUALINC1/P/B/Q/QNS/BQ/BQNS address patterns, the INDEXINC register defines the amount to increment each subsequent address by for the second of two increment sequences.

HALT Register

The 1-bit HALT register (D4) halts the current addressing immediately when set HIGH. If halted, the inherent seven SYSCLK cycle latency will occur after the next START signal and before the first address is generated.

DIGITREV Register

The 20-bit (two 8-bit and one 4-bit register) DIGITREV register (D5-D7) is used to calculate the digit reverse sequence for the first column of an FFT.

For DSP24 based FFT's, the DIGITREV value is 0.

For non DSP24 implementations it is determined based on the product of the radix combination. For example, an 8-point FFT could contain three different digit reverse sequences depending on the radix combination being used as follows:

Radix-2 x Radix-2 x Radix x 2

Radix-2 x Radix-4

Radix-4 x Radix-2

Note: DIGITREV is the only register that requires the user to provide information concerning the entire algorithm being performed, rather than a specified pass.

MEMSIZE Register

The 21-bit (two 8-bit and one 5-bit register) MEMSIZE register (D8-DA) is used by the following address patterns:

TF2Cn (0-19) TF4Cn (0-18), TF8Cn (0-17), TF16Cn (0-16), TF32Cn (0-15), TF2n (0-19) TF4n (0-9), TF16n (0-4), MXT24n (0-8), MXT216n (0-3), MXT416n (0-3), MXT2416n (0-3), DECIM, ADECIM, BFCTT, DUALINC1Q, DUALINC1QNS, DUALINC1BQ, DUALINC1BQNS, DUALINC2Q, DUALINC2QNS, DUALINC2BQ, and DUALINC2BQNS.

For these address patterns, MEMSIZE defines the physical memory size that contains 360 degrees of coefficient data for address patterns on the coefficient side of the execution unit. MEMSIZE permits a transform to be completed using a coefficient array of any size when MEMSIZE is greater or equal to N, and both MEMSIZE and N are a power of 2.

When used in conjunction with MODE bit 4 (DEG90), the table. When MODE bit-4 is set active HIGH, the addresses that are generated will come from the 0 to MEMSIZE/4-1 address range.

CHIPID Register

This 8-bit, read only register (DB) contains chip revision identification:

Bit[3:0]Last digit for the year

Bit[7:4]Number for the month

For example,

CHIPID = 67H would indicate that the chip version was updated during June, 1997.

INCADRSTART Register

The 20-bit (two 8-bit and one 4-bit register) INCADRSTART register (DF-E1) is used by the DUALINC1/P/B/Q/QNS/BQ/BQNS address patterns to increment the current ADRSTART value for the first of the two increment sequences.

The current ADRSTART value is updated after each pair of (ADR and INDEX) address increment patterns executes.

INCINDEXSTART Register

The 20-bit (two 8-bit and one 4-bit register) INCINDEXSTART register (E2-E4) is used by the DUALINC1/P/B/Q/QNS/BQ/BQNS address patterns to increment the current INDEXSTART value for the second of the two increment sequences.

The current INDEXSTART value is updated after each pair of (ADR and INDEX) address increment patterns executes.

INCADR2START Register

The 20-bit (two 8-bit and one 4-bit register) INCADR2START register (E5-E7) is used by the DUALINC2/P/B/Q/QNS/BQ/BQNS address patterns to increment the current ADR2START value for the first of the two increment sequences.

The current ADR2START value is updated after each pair of (ADR and INDEX) address increment patterns executes.

INCINDEX2START Register

The 20-bit (two 8-bit and one 4-bit register) INCINDEX2START register (E8-EA) is used by the DUALINC2/P/B/Q/QNS/BQ/BQNS address patterns to increment the current INDEX2START value for the second of the two increment sequences.

The current INDEX2START value is updated after each pair of (ADR and INDEX) address increment patterns executes.

MASTEROFFSET Register

The 20-bit (two 8-bit and one 4-bit register) MASTEROFFSET register (EB-ED) is used to add an arbitrary 20-bit value to each and every address that is generated by the current address pattern and associated configuration registers. See Figure 6.

arbitrary 20-bit value to each and every address that is generated by the current address pattern and associated configuration registers.

MASTEROFFINC Register

The 20-bit (two 8-bit and one 4-bit register) MASTEROFFINC register (EE-F0) is used increment the current value of the MASTEROFFSET register after each successive address pattern is finished, for each count of the MASTERREPEAT register greater than zero.

MASTERREPEAT Register

The 20-bit (two 8-bit and one 4-bit register) MASTERREPEAT register (F1-F3) is divided into two fields.

The first field (bits 18-0), is used to define how many additional times the MMU24 will execute another address sequence. (A value of 2 indicates that the MMU24 will run 3 complete sequences.)

The second field/MSB, is used to defines what successive address sequences will be.

When set LOW, successive address patterns will be repeats of the current pattern pointed to by the Program Counter and its' associated configuration registers.

When set HIGH, successive address patterns will come from successive locations in Program Memory, and their associated configuration registers. Thus concatenating multiple sequences together to form ever more complex patterns.

The functionality of the MASTERREPEAT register, is to internally pre-START the MMU24, (the same as double strobing the START signal externally), with the additional feature that the TC signal covers the whole concatenated pass. That is it does not strobe between sequences like an externally strobed START pulse would cause.

Note: If the user would like a version of TC that DOES pulse between sequences, the TCP signal's REPEAT bit in the SKEW register may be set to accomplish this.

SKEW Register

The 22-bit (two 8-bit and one 6-bit register) SKEW register (F4-F6) is used to move the timing of various signals forward and backward an integer number of cycles relative to the address output. The bits for the SKEW register shown in Figure 4 are defined as follows:

Bits 2-0, Determines the relative skew of the $\overline{\text{MEMOE}}$ signal. The $\overline{\text{MEMOE}}$ signal is delayed by an additional number of SYSCLK cycles determined by the unsigned value of these bits. (except for the value of 7 which corresponds to a negative delay of one cycle) See Figure 4.

Bits 5-3, Determines the relative skew of the $\overline{\text{MEMW}}$ signal. The $\overline{\text{MEMW}}$ signal is delayed by an additional number of SYSCLK cycles determined by the unsigned value of these bits. (except for the value of 7 which corresponds to a negative delay of one cycle)

Bits 8-6, Determines the relative skew of the CCOMR, CCOMI, and SWAP signals. The all of these signals are delayed by an additional number of SYSCLK cycles determined by the unsigned value of these bits. (except for the value of 7 which corresponds to ZERO delay cycles. Same as a value of 0.)

Bits 11-9, Determines the relative skew of the TC signal. The TC signal is delayed by an additional number of SYSCLK cycles determined by the unsigned value of these bits. (except for the value of 7 which corresponds to a negative delay of one cycle)

Bits 14-12, Determines the relative skew of the TCP signal. The TCP signal is delayed by an additional number of SYSCLK cycles determined by the unsigned value of these bits. (except for the value of 7 which corresponds to a negative delay of one cycle)

Bit 15, Determines whether the TCP signals rising edge will be delayed by an additional SYSCLK cycle. This bit the equivalent of MODE bit-3 for the TC signal.

Bit 16, Sets the polarity of the TCP signal. When set LOW, TCP is active LOW, when set HIGH, TCP is active HIGH.

Note: TCP will always power up inactive HIGH.

Bit 17, Determines whether or not the TCP signal will strobe inactive between successive address sequences initiated by the MASTERREPEAT register.

When set LOW, TCP will act like TC. It will go active with the first address out, and inactive one cycle before the last address of the last repeated sequence. (Plus or minus SKEW and DELAY settings.)

When set HIGH, TCP will pulse inactive one cycle before the last address of each repeated address sequence. (Plus or minus SKEW and DELAY settings.)

Bits 20-18, Determines the relative skew of the PO signal. The PO signal is delayed by an additional number of SYSCLK cycles determined by the unsigned value of these bits. (except for the value of 7 which corresponds to a negative delay of one cycle).

Bit 21, Sets the polarity of the PO signal for the DUALINC1P and DUALINC2P address patterns.

When set LOW, the PO signal will go active during the first address sequence of each set of address sequence pairs that are executed (i.e. each ADRSTART/ADRLENGTH/ADRINC sequence.)

When set HIGH, the PO signal will go active during the second address sequence of each set of address sequence pairs that are executed (i.e. each INDEXSTART/INDEXLENGTH/INDEXINC sequence.)

Figure 6. Internal Memory Description

ADDRESS	BITS USED	NAME	VALID RANGE (HEX)*	DESCRIPTION
0-1F	8	PROGRAM MEMORY	00-FF	32, 8 bit register locations for storing address patterns. The Program Counter determines which is used.
20-3F	8	LATENCY MEMORY	00-FF	32, 8 bit register locations for storing latency values (bits [6:0]). Bit 7 indicates reading or writing the SRAM The Program Counter determines which is used.
40-9F	9	EXTENDED FEATURE MEMORY	000-1FF	32, 9 bit register locations reserved to store additional latency values (bits [4:0]). Skew range for the $\overline{\text{ACTIVE}}$ signal (bit [7:5]) and an enable for ACTIVE (bit 8)
A0	5	PCSTART	00-1F	For internal memory mode, loads the Program Counter with the first index to be executed from Program/Latency/ Feature memory (disabled in external memory mode).
A2	6	MODE	00-7F	Defines the various modes of operation
A3-A5	20	OVERLAP/ DINC2PAIRS	00000-FFFFF	OVERLAP: Defines the amount of overlap for the data. DINC2PAIRS: Defines the number of ADR-INDEX pairs of sequences for the DUALINC2 functions.
A6-A8	20	PAUSE	00000-FFFFF	Defines the number of cycles to extend the last address and control signals output, before starting next function.
A9-AB	20	BREAKPOINT	00000-FFFFF	Active when MODE register bit1 is set HIGH. Causes the MMU24 to stop generating addresses after the number of addresses generated in the sequence is equal to the value in the BREAKPOINT register.
AC-AE	21	N	000000-100000	For most address patterns, N defines the length of the address sequence, except those that use 2N, ADRLENGTH, or INDEXLENGTH.
AF-B1	20	ADR2START	00000-FFFFF	Defines the starting address location for the first half sequence of the DUALINC2 pattern.
B2-B4	20	DFACTOR/ ADR2LENGTH	00000-FFFFF	DECIM: Defines the value that each address is to be incremented by each time an address is generated. ADECIM: Uses the value as in DECIM, but only for the first addresses. DUALINC2: Uses the value as the address length of the 1st half sequence.
B5-B7	20	LEAP/ADR2INC	00000-FFFFF	ADECIM: Defines the accumulation amount for the DFACTOR value. DUALINC2: Uses the value as the address increment for the 1st half sequence.
B8-BA	20	NLEAP /INDEX2LENGTH	00000-FFFFF	ADECIM: Defines the number of sets of N addresses that are to be generated. DUALINC2: Uses the value as the address length for the 2nd half sequence.
BB-BD	20	INDEX2START	00000-FFFFF	DUALINC2: Defines the starting address for the 2nd half sequence.
BE-CO	20	ZEROPAD/ INDEX2INC	00000-FFFFF	PADLOW/P/EP/LP, PADHIGH/P/EP/LP: Defines the number of INDEXing addresses to be generated before or after the ADR addressing starts. DUALINC2: Defines the address increment for the 2nd half sequence.
C1-C3	20	ADRSTART	00000-FFFFF	PADLOW/P/EP/LP, PADHIGH/P/EP/LP, MODINC, MODDEC, and INC: Defines the starting address in the current sequence. DUALINC1: Defines the starting address of the 1st half sequence.
C4-C6	20	ADRLENGTH	00004-100000	PADLOW/P/EP/LP, PADHIGH/P/EP/LP, MODINC, MODDEC, and INC: Indicates the number of points in the current sequence. DUALINC1: Defines the number of addresses of the 1st half sequence
C7-C9	20	ADRINC	00000-FFFFF	PADLOW/P/EP/LP, PADHIGH/P/EP/LP, MODINC, and INC: Indicates the amount each subsequent address is to be incremented by, when generated. DUALINC1: Defines the address increment of the 1st half sequence

Figure 6. Internal Memory Description (cont)

ADDRESS	BITS USED	NAME	VALID RANGE (HEX)*	DESCRIPTION
CA-CC	20	ADRDEC/ INDEXLENGTH	00000-FFFFF	MODDEC: Indicates the amount each subsequent address pattern is to be decrement. DUALINC1: Defines the number of addresses of the 2nd half sequence
CD-CF	20	INDEXSTART	00000-FFFFF	PADLOW/P/EP/LP, PADHIGH/P/EP/LP, DISCARD/P/EP/LP, BFCTU/P/EP/LP: Points to an address space outside of the working space memory for a particular address pattern. DUALINC1: Defines the stating address of the 2nd half sequence
D0-D2	20	INDEXINC	00000-FFFFF	PADLOW/P/EP/LP, PADHIGH/P/EP/LP, DISCARD/P/EP/LP: Identifies the amount to increment the INDEXSTART pointer value by each time an indexing address is generated. DUALINC1: Defines the address increment of the 2nd half sequence.
D3	5	PCEND	00-1F	For internal memory mode, points to the last address pattern to be executed in the program memory before loading PCSTART. (disabled in external memory mode).
D4	1	HALT	0-1	Halts the current addressing sequence after BREAKPOINT address have been generated.
D5-D7	20	DIGITREV	00000-FFFFF	Defines the digit reverse sequence for the first column of an FFT based on the combination of the radices used.
D8-DA	21	MEMSIZE	000000-100000	TFx, DECIM, ADECIM, BFCTT: Defines the physical memory size that contains 360 degrees of coefficients in a COS/SIN table (Whether or not they are all actually stored.)
DB	8	CHIPID	67	Chip identification/revision. (Read only)
DC-DE	19	DISCARD/ DINC1PAIRS	00000-FFFFF	DISCARD/P/EP/LP: Defines the number of addresses to be discarded. DINC1PAIRS: Defines the number of ADR-INDEX pairs of sequences for the DUALINC1 functions
DF-E1	19	INCADRSTART	00000-FFFFF	Defines the amount to increment the first starting address by (ADRSTART), after executing both sequences of a ADR/INDEX DUALINC1 pair.
E2-E4	19	INCINDEXSTART	00000-FFFFF	Defines the amount to increment the second starting address by (INDEXSTART), after executing both sequences of a ADR/INDEX DUALINC1 pair.
E5-E7	19	INCADR2START	00000-FFFFF	Defines the amount to increment the first starting address by (ADR2START), after executing both sequences of a ADR/INDEX DUALINC2 pair.
E8-EA	19	INCINDEX2START	00000-FFFFF	Defines the amount to increment the second starting address by (INDEX2START), after executing both sequences of a ADR/INDEX DUALINC2 pair.
EB-ED	19	MASTEROFFSET	00000-FFFFF	Defines a value (offset) to add to all addresses generated by all address patterns defined by the above registers.
EE-F0	19	MASTEROFFINC	00000-FFFFF	Defines a value to increment the MASTEROFFSET value by, after each iteration defined by the MASTERREPEAT.
F1-F3	19	MASTERREPEAT	00000-FFFFF	Bits [18:0] define the number additional times to execute an address pattern. (Performs an internal double strobe of the START signal.) Bit 19 when set HIGH will cause the Program Counter to increment on successive (repeated) address patterns, which will essentially concatenate multiple address patterns.
F4-F6	22	SKEW	000000-7FFFFF	Defines positive and negative cycle timing of various output control signals relative to ADDR[19:0], measured in multiples of SYSCLK.

Table 4. Host Decoder Truth Table

A1	A0	MODE(0)=L Internal Memory Programming	MODE(0)=H External Memory Programming
L	L	Data	Location 31 of Program Memory (3F)
L	H	Address	Location 31 of Program Memory (3F)
H	L	Reserved	Mode register (A2)
H	H	Reserved	Reserved

HOST - INTERFACE

The host interface enables the MMU24 to be programmed in internal or external memory mode via the 8-bit data bus DB[7:0] as was shown in Figure 2. In particular, the host decode enables the data to be written to a specified address location depending on the A1, A0 signals and whether it is in internal or external mode as shown in Table 4.

Internal Memory Mode

For internal memory mode, the A1 and A0 signals determine whether the DB[7:0] bus is latched into an address register (A1 is LOW and A0 is HIGH) or one of many data registers (A1 is LOW and A0 is LOW). In internal mode, the address register is decoded into an array of enable signals that determine which specific data register is accessed for subsequent data reads and data writes. Thus, the host decode loads the MMU24 registers through a two step scheme: registering of the desired address by writing to DB[7:0] with A1 LOW and A0 HIGH, and then reading or writing data to the register just selected by reading/writing DB[7:0] with A1 LOW and A0 LOW. The value of the DIR signal determines whether the DB[7:0] bus is in input mode for writing (DIR set LOW) or if the DB[7:0] bus is in output mode for reading registers (DIR set HIGH).

External Memory Mode

In external memory mode, A1 and A0 permit the DB[7:0] data bus to flow to the following predefined address locations:

- Location 31 of Program Memory (1F) - A1 is LOW and A0 is LOW
- Location 31 of Latency Memory (3F) - A1 is LOW and A0 is HIGH
- Location A2, the MODE register (A2) - A1 is HIGH and A0 is LOW

In this mode, the address register is bypassed and the DB[7:0] data bus transfers data to the predefined address (1F, 3F, or A2) as assigned by the A1 and A0 signals. In addition, the MODE register address location (A2) can be accessed externally to toggle back and forth between the internal and external memory modes. See Table 4.

Note: *because the address register is bypassed, programming in external memory mode requires only half the program length that would be utilized in internal memory mode, once the configuration registers have been programmed.*

ADDRESS CALCULATION

The address calculation logic of the MMU24, produces address patterns using the following logic:

- Opcode Decode
- Sequence Generation
- Quadrant Lookup
- Master Offset and Repeat
- Control Signal Generation

The address calculation logic decodes each address pattern and selects a set of seeds values which, in turn, control the actual address sequence generation. The generated sequence of addresses then pass through the quadrant logic which can reduce the required coefficient table by a factor of four when enabled.

After the Quadrant logic, each address passes though the Master Offset logic, where the current Master Offset value is added on to generate the final output address.

The progress of address generation, and the SKEW of individual control signals is performed in parallel with the address generation in the Control Signal Generation logic.

To help solve testability issues, the address calculation logic contains a signature analyzer at the output of the final address, that includes a primitive polynomial of degree 31. The signals that are connected to the signature analyzer are as follows:

D[19:0] = ADDR[19:0]
D[25] = CCOMI_EN (internal)
D[20] = PO
D[26] = SWAP
D[21] = PO_EN (internal)
D[27] = SWAP_EN (internal)
D[22] = CCOMR
D[28] = NOT TC
D[23] = CCOMR_EN (internal)
D[29] = NOT ACTIVE
D[24] = CCOMI
D[30] = MEMW and MEMOE

Note: the MMU24 contains two instructions (opcodes) that are reserved for clearing and viewing the signature analyzer via the 20-bit address output. CLRSIG is the instruction for clearing the signature to zero. (The signature is always cleared after a reset.) VIEWSIG is the instruction that displays the signature contents since the last CLRSIG instruction.

In addition to the CLRSIG and VIEWSIG instructions, the MMU24 allows the user to control these operations via two bits in the MODE register. MODE bit-7 clears the signature register like the CLRSIG instruction. MODE bit-6 functions like the VIEWSIG instruction by enabling the output of the signature register on ADDR[19:0], when set HIGH, and re-enabling the current address output on ADDR[19:0] when set LOW.

Opcode Decode

The current opcode consists of the current address pattern, Latency value, Read/Write value, and $\overline{\text{ACTIVE}}$ control signals currently being pointed to by the internal Program Counter. These values are decoded in a straightforward manner to select a particular set of seeds for each address pattern.

These seeds are selected from the Configuration Registers and various constants. They are selected according to the address pattern and determine everything about the desired address pattern to be generated. (i.e. They set algorithms, loop sizes, initial values, increment amounts, etc.)

Sequence Generation

The complexity of the Sequence Generation logic is set up to perform the radix-32 algorithm. All other address patterns are generated using subsets.

The Sequence Generation logic is composed of a series of adders that perform arithmetic operations and manipulate addresses for a specified algorithm.

Quadrant Lookup

The Quadrant Lookup logic takes the address out of the Sequence Generation block and compares it to the range of the lookup table defined by the MEMSIZE register. When enabled, the Quadrant Lookup logic then outputs a set of CCOMR, CCOMI, and SWAP signals for each address, based on which quadrant of the table the address is in, and then forces the address to point into the first quadrant to save memory.

Master Offset and Repeat

The Master Offset logic is composed of two main blocks, the offset/increment block, and the repeat block. Together, they work to add an additional dimension to the address generation, by allowing the user to add arbitrary offsets to all generated address patterns, and to seamlessly concatenate successive patterns together to build arbitrarily complex new unique patterns. See Figure 5.

The offset block is composed of a address offset adder and an adder to increment the offset. On the very first address of the very first sequence (if there are any repeats), the offset register is loaded with the MASTEROFFSET register as the initial value to offset the incoming addresses by. After each sequence of done, if the user has chosen to repeat/concatenate successive sequences, a new offset is set from the value of the current offset plus the MASTEROFFINC register value.

The repeat block controls the selection of the offsets, and the registering of new offsets on a pass by pass basis. Additionally, the repeat logic signals the MMU24 to start a new address sequence seamlessly from either the current Program Counter location, or the next Program Counter location, by incrementing it.

Control Signal Generation

The Control Signal Generation logic generates all the flags and memory control signals output from the MMU24 with the relative cycle timing to ADR[19:0] defined by the MODE, SKEW, and EXTENDED FEATURE Memory registers.

CLOCK CONTROL

The clock control logic detects the rising edge of the START signal, which initiates the execution of the current address pattern pointed to by the Program Counter. The Program Counter is incremented each time a rising edge of the START signal is detected.

PROGRAM COUNTER

The Program Counter is a 5-bit counter whose starting address is given by the PCSTART register and ending address is given by the PCEND register. The Program Counter is incremented by one each time the rising edge of the START signal is detected. Upon reaching a value equal to PCEND, the Program Counter is again re-loaded with the value in the PCSTART register. The Program counter will automatically be loaded from the PCSTART register after reset or when a change in the PCSTART register is detected.

Note: For external memory mode, PCSTART and PCEND are bypassed. The data is transferred directly to the predefined addresses (1F, 3F, or A2) depending on the value of the A1, A0 signals.

USING THE MMU24

Initializing the MMU24

The MMU24 is initialized when the RESET signal is set active HIGH. When this occurs, all internal counters and registers are reset to zero. The RESET signal is asynchronous and must be held HIGH for the minimum pulse duration for all registers to be reset correctly.

After the MMU24 is reset, all internal execution and configuration memory registers must be loaded with appropriate values, if different than there initial conditions. Only the execution memory that is used needs to be programmed, and only the configuration registers associated with each programmed address pattern need be programmed. (See Figure 9., the Address Pattern/Register Cross Reference.)

The MMU24 uses a simple "Tiered - Decode" that allows it to be programmed like a common peripheral device off an 8-bit address/data bus. When used with the DSP24, the DSP24's Scheduler can program a group of MMU24's by selecting individual/multiple MMU24's with their chip select (CS) lines.

The Tiered - Decode means that the user must separately program address and data values. To write internal registers with data, the user must first write the 8-bit location/address of the target register. Then the user can either write the data value, or read the current registers contents.

Starting Execution

The generation of address sequences begins when the rising edge of the START signal is detected. Before this occurs, all relevant Program Memory, Latency Memory, Extended Feature Memory, and Configuration registers must be loaded with valid values. For the first START detected after a RESET, or for START's that occur after the MMU24 as finished its current sequence, the MMU24 takes 8 rising edges of the SYSCLK (approximately 7 cycles) before the first address is output. However, if a rising edge of the START signal is detected during a running address sequence, and occurs more than 6 cycles before last address, then when the MMU24 will "pre-start" the next address sequence immediately after the current address pattern finishes, without any pipeline delay.

Note: Since "Pre-Starting" the MMU24 by strobing the START signal during a pass is equivalent to setting the MASTERREPEAT register to a non-zero value, Pre-Starting should not be used if the MASTERREPEAT register is being used.

Stopping Execution

The MMU24 is a deterministic machine. Given any set of initial conditions it will eventually finish generating address outputs and stay at the last generated output. The MMU24 then enters a waiting state. After waiting for the number of cycles in the PAUSE register, the MMU24 will be ready to start new address patterns.

In order to pre-maturely stop the current sequence of addresses or pause cycles, two methods may be used. The first is to set the HALT register bit-0 HIGH to immediately stop the current sequence. The second is to set the MODE register bit-1 HIGH to enable the BREAKPOINT counter to count addresses and force and end after the number of addresses generated equals the value in the BREAKPOINT register.

Pausing Execution

The generation of new addresses output from the MMU24 may be paused by setting the $\overline{\text{SYSCLKEN}}$ signal HIGH for as many SYSCLK cycles as are desired.

PROGRAMMING THE MMU24

The MMU24 can be programmed with a wide variety of hardware. Examples include the DSP24 Scheduler, an FPGA scheduler derived from the FPGA Scheduler Design Kit, or custom implementations able to write peripheral devices with a two step address then data write sequence.

Interface Between the MMU24 and the Scheduler

Programming the MMU24 is composed of two separate accesses. The first access, writes the address of the desired register by setting the $\overline{\text{CS}}$ signal LOW, the DIR signal LOW, the A1 signal LOW, the A1 signal HIGH and strobe the R/$\overline{\text{W}}$ signal LOW and then HIGH to register the address written. Once written the user programs the data register with the desired value by setting the $\overline{\text{CS}}$ signal LOW, the DIR signal LOW, the A1 signal LOW, the A1 signal LOW and strobe the R/$\overline{\text{W}}$ signal LOW and then HIGH to register the data to be written.

Data may be read from the MMU24 using a similar method. The first access, writes the address of the desired register by setting the $\overline{\text{CS}}$ signal LOW, the DIR signal LOW, the A1 signal LOW, the A1 signal HIGH and strobe the R/$\overline{\text{W}}$ signal LOW and then HIGH to register the address written. Once written the user reads the data register value by setting the $\overline{\text{CS}}$ signal

LOW, the DIR signal HIGH, the A1 signal LOW, the A1 signal LOW and strobe the R/$\overline{W}$ signal LOW and then HIGH to read the registers value.

When set up in a target system with the DSP24, each MMU24 will be controlled by the scheduler via separate $\overline{CS}$ signals, and will in turn communicate their own status by each sending the TC signal back to the Scheduler.

By sharing common R/$\overline{W}$, DIR, A1, A0, and DB[7:0] signals, the Scheduler is able to write the MMU24's individually using the separate $\overline{CS}$ line, or to "gang" program them if multiple MMU24's need the same register programmed with the same value.

The MMU24 operates in either Internal or External programming modes. If configuration registers need to be programmed, then the Internal Programming mode must be used. When using this mode, the user has access to all the MMU24's internal memory, which includes up to 32 Program Memory locations for storing address patterns to be executed. If the user has algorithms that are longer then 32 instructions long, or if the user is using multiple algorithms that together make up more than 32 instructions, then the user can use the External Programming mode.

In External Programming mode, the user is free to issue an infinite number of instructions to the MMU24. However, the only configuration register that may be programmed is the MODE register. If configuration registers need to be programmed and/or changed, then the user must program the MODE to return to Internal Programming mode, so the registers can be programmed, and then re-return to External Programming mode by re-programming the MODE register.

Figure 7. Master Offset and Repeat Register

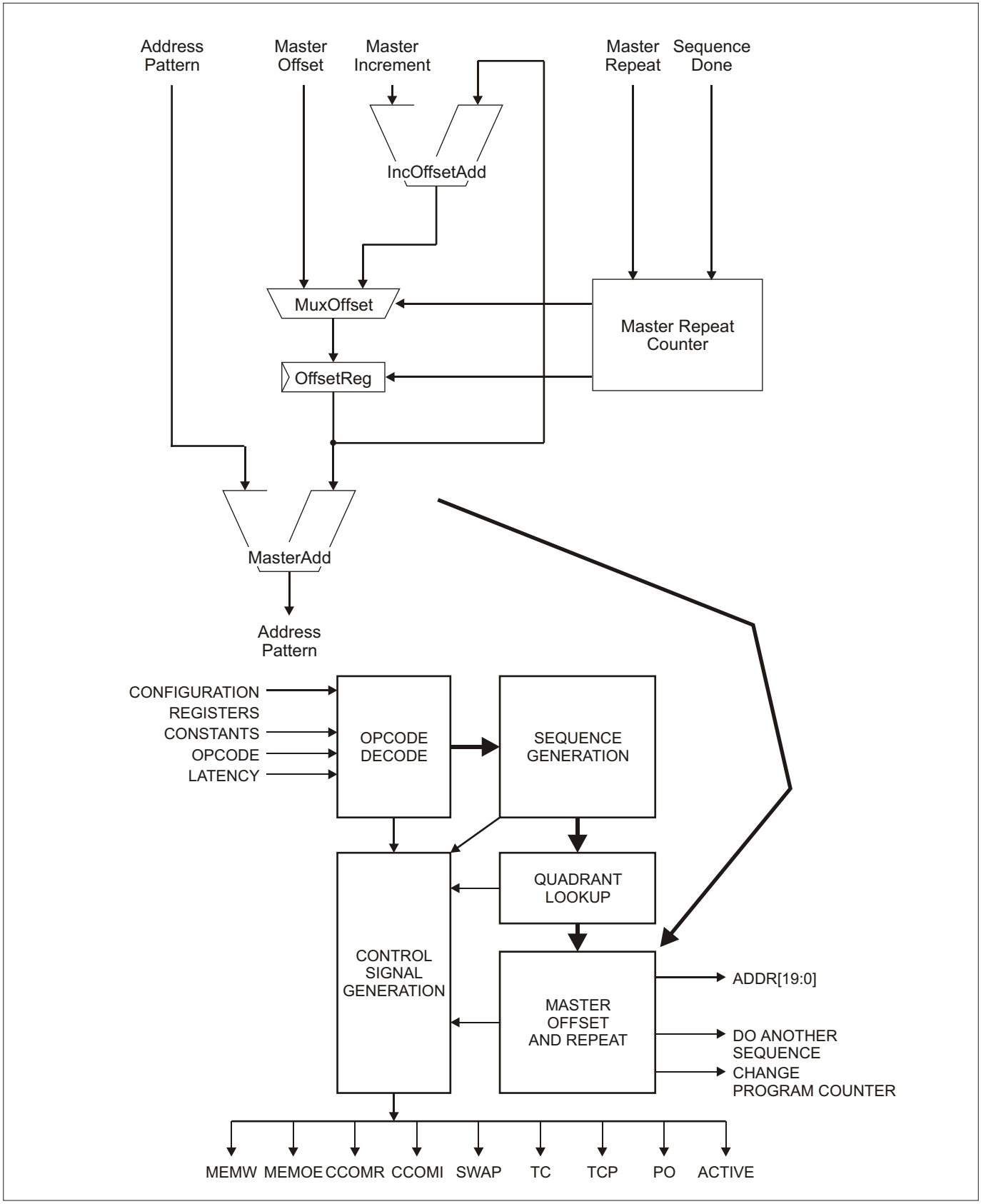
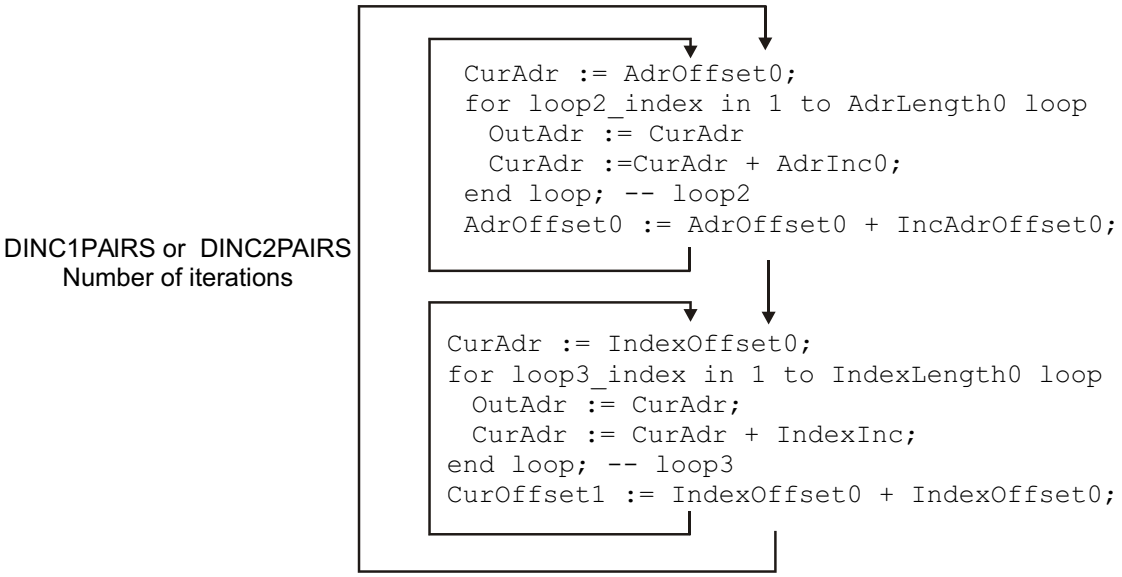


Figure 8. DUALINC Patterns



CUSTOM ADDRESS PATTERNS

The DUALINCx address patterns are an extension of the INC address pattern. The only difference between DUALINC1x pattern and the DUALINC2x pattern is that they are defined by a different set of configuration registers as seen in Figure 9, the Pattern Register Cross Reference.

The DUALINCx address patterns allow the user to embed two separate INC type of address patterns, defined with ADRSTART, ADRLENGTH, and ADRINC for the first INC pattern and INDEXSTART, ADRDEC_INDEXLENGTH, INDEXINC for the second address pattern, within a global loop. This overriding loop then controls the number, equal to DINC1PAIRS, of iterations that a combined, the first INC followed by the second INC, address pattern is executed. In addition, the global loop defines a pair of starting address increments that are to be added to the starting address of each INC pattern after each pair executes. The INCADRSTART register adds offset onto the ADRSTART initial value, and the INCINDEXSTART adds offset onto the INDEXSTART initial value for each iteration. See Figure 8.

As an example lets perform the following:

Define A as the first INC type sequence. If we set ADRSTART, ADRLENGTH, and ADRINC to 1, 5, and 7, then A's address pattern is defined as 01, 08, 0F, 16, 1D.

Define B as the second INC type sequence. If we set INDEXSTART, INDEXLENGTH, and INDEXINC to 9, B, and D, then B's address pattern is defined as 09, 16, 23, 30, 3D, 4A, 57, 64, 71, 7E, 8B.

Now, when the global loop is defined to execute 3 (DISCARD_DINC1PAIRS = 3) times, thus we get the following sequence:

← Sequence A	← Sequence B
00001 00008 0000F 00016 0001D	00009 00016 00023
00030 0003D 0004A 00057 00064	00071 0007E 0008B
00001 00008 0000F 00016 0001D	00009 00016 00023
00030 0003D 0004A 00057 00064	00071 0007E 0008B
00001 00008 0000F 00016 0001D	00009 00016 00023
00030 0003D 0004A 00057 00064	00071 0007E 0008B

i.e. Sequence A followed by Sequence B, repeated 3 times.

Then if we set the increment for the starting offsets to INCADRSTART = 2 and INCINDEXSTART = 4, they were set to zero above, the sequence becomes:

Sequence A	Sequence B
00001 00008 0000F 00016 0001D	00009 00016 00023
00030 0003D 0004A 00057 00064	00071 0007E 0008B
00003 0000A 00011 00018 0001F	0000D 0001A 00027
00034 00041 0004E 0005B 00068	00075 00082 0008F
00005 0000C 00013 0001A 00021	00011 0001E 0002B
00038 00045 00052 0005F 0006C	00079 00086 00093

Simplified Mixed Radix FFT Addressing

The MMU24 generates each FFT addressing sequence using an address sequence for each column of the decomposed FFT, i.e. one sequence for each access to the memory holding the sampled array of data.
For example a million point complex FFT can be performed several ways:

10 radix-4 address patterns
4 radix-32 address patterns
2 radix-1024 address patterns

The MMU24 can generate FFT twiddle and data I/O addresses at any column of an algorithm made up of radix-2, radix-4, radix-8, radix-16, and radix-32.
Because the addressing depends on where you are in the execution of the FFT algorithm, the opcode controlling the address pattern to be generated is a combination of which radix the user is currently processing and how far into the algorithm the current data is.
The measure used for determining how far the current algorithm has progressed, is the radix-2. Each of the higher radices can be thought of as consisting of multiple radix-2 equivalents. By counting the number of equivalents used previously, the user can determine the appropriate opcode to use.

1024 Point FFT Example:

Using Table 5 select the address pattern mnemonics required to generate the data addresses using a mixed radix combination of radix-4, radix-16, and radix-16, (i.e. 4x16x16 =1024).
Table 5 is a conversion table which indicates the data address to use for each column of a mixed radix FFT.

1. Write the mixed radix structure of the FFT, starting from the left as indicated.

Mixed radix columns:

0 1 2

R4 x R16 x R16

2. Convert to radix-2 equalivents :

R4 x R16 x R16

R2xR2, R2xR2xR2xR2, R2xR2xR2xR2

3. Successively number the starting R2 equivalent for each column:

Table 5. Opcode Selection for Executing Transforms

COLUMN	INPUT/ OUTPUT	Coefficient				
		RADIX-2	RADIX-4	RADIX-8	RADIX-16	RADIX-32
0	BFC0	TF2C0	TF4C0	TF8C0	TF16C0	TF32C0
1	BFC1	TF2C1	TF4C1	TF8C1	TF16C1	TF32C1
2	BFC2	TF2C2	TF4C2	TF8C2	TF16C2	TF32C2
3	BFC3	TF2C3	TF4C3	TF8C3	TF16C3	TF32C3
4	BFC4	TF2C4	TF4C4	TF8C4	TF16C4	TF32C4
5	BFC5	TF2C5	TF4C5	TF8C5	TF16C5	TF32C5
6	BFC6	TF2C6	TF4C6	TF8C6	TF16C6	TF32C6
7	BFC7	TF2C7	TF4C7	TF8C7	TF16C7	TF32C7
8	BFC8	TF2C8	TF4C8	TF8C8	TF16C8	TF32C8
9	BFC9	TF2C9	TF4C9	TF8C9	TF16C9	TF32C9
10	BFC10	TF2C10	TF4C10	TF8C10	TF16C10	TF32C10
11	BFC11	TF2C11	TF4C11	TF8C11	TF16C11	TF32C11
12	BFC12	TF2C12	TF4C12	TF8C12	TF16C12	TF32C12
13	BFC13	TF2C13	TF4C13	TF8C13	TF16C13	TF32C13
14	BFC14	TF2C14	TF4C14	TF8C14	TF16C14	TF32C14
15	BFC15	TF2C15	TF4C15	TF8C15	TF16C15	TF32C15
16	BFC16	TF2C16	TF4C16	TF8C16	TF16C16	XXXXXX
17	BFC17	TF2C17	TF4C17	TF8C17	XXXXXX	XXXXXX
18	BFC18	TF2C18	TF4C18	XXXXXX	XXXXXX	XXXXXX
19	BFC19	TF2C19	XXXXXX	XXXXXX	XXXXXX	XXXXXX

R2xR2, R2xR2xR2xR2, R2xR2xR2xR2
0 1 2

4. Using Table 5 extract the corresponding opcode:

BFC0/TF4C0, BFC2/TF16C2, BFC6/TF16C6

Figure 9. MMU24 Address Pattern/Register Cross Reference

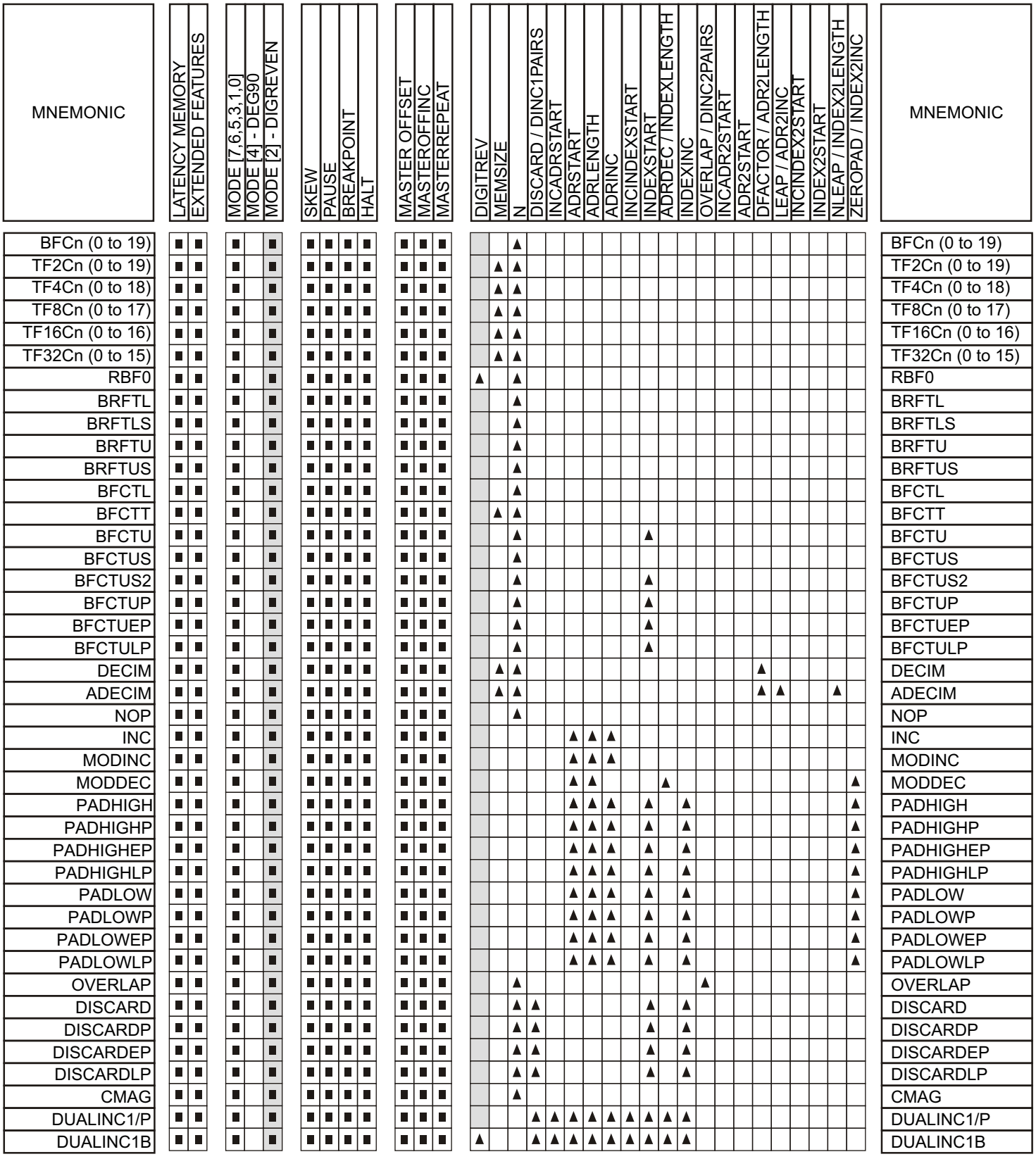
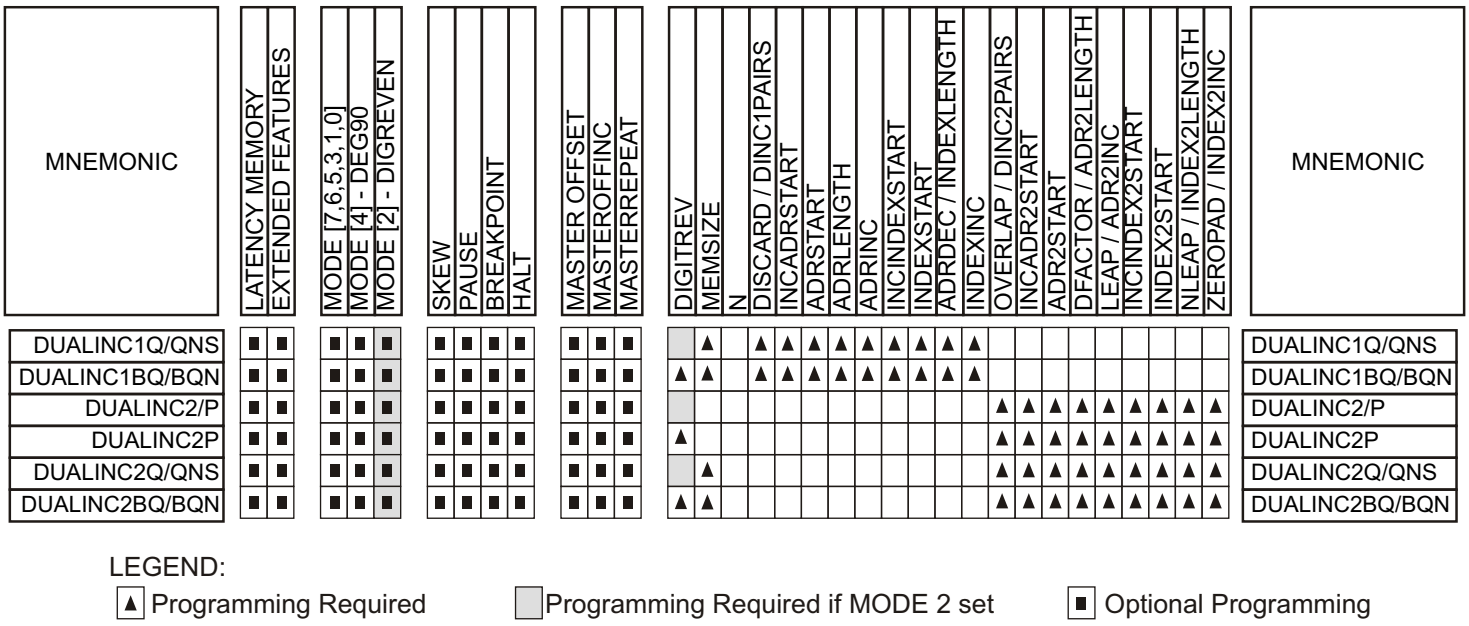


Figure 9. MMU24 Address Pattern/Register Cross Reference (cont.)



LEGEND:

▲ Programming Required Programming Required if MODE 2 set ■ Optional Programming

ABSOLUTE MAXIMUM RATINGS [1,2]

Supply Voltage to VSS Potential	-0.3 V to 5V
Signal Pin Voltage to VSS Potential	-0.3 to VDD +0.3 V (not to exceed 5 V)
DC Output Current	± 10 mA
Storage Temperature range	-40 deg C to 125 deg C
Power Dissipation (Package Limit)	2.0 W
DC Voltage Applied to Outputs in High-Z State	-0.3 to VDD +0.3 V (not to exceed 5 V)

OPERATING RANGE [1]

		MIN	MAX	UNIT
TA	Temperature, Ambient	0	70	deg C
VDD	Supply Voltage	3.13	3.43	V
VSS	Supply Voltage	0.0	0.0	V
VIL	Logic '0' Input Voltage [5]		0.3 VDD	V
VIH	Logic '1' Input High Voltage	0.7 VDD	VDD+0.3	V

DC ELECTRICAL CHARACTERISTICS [1] (Over Operating Range)

SYMBOL	DESCRIPTION	TEST CONDITIONS	MIN	MAX	UNIT
I_{LPUI} 6	Current	VDD=3.13 V, VIN=0 V	-70	-15	uA
		VIN=VDD	-10	10	
I_{LPDI} 7	Pull-Down Input Leakage	VDD=3.13, VIN=VDD	15	70	
		VIN=VSS	-10	10	
I_{LI}	Input Leakage Current	VDD=3.13 V, VIN= 0 to VDD	-10	10	
V_{LOH}	Output High Voltage	IOH=-6.0 mA	1.58		V
V_{OL}	Output Low Voltage	IOL=6.0 mA		0.26	V
I_{DDI}	Average Supply Current	Measured at tcyc (100MHz)		250	mA
I_{DD2}	Average Standby Current	All inputs = VIH (0.3 VDD), 2 SYSCLKS occur to enable RESET of CCOMR, CCOMI, & SWAP. Excludes output load current.		20	mA
I_{DD3}	Quiescent Standby Current	All inputs = VIH (0.3 VDD), 2 SYSCLKS occur to enable RESET of CCOMR, CCOMI, & SWAP. Excludes output load current.		1	mA

See notes on Page 30

AC TEST CONDITIONS [1]

PARAMETER	RATING
Input Pulse levels	VSS to 3.0 VDD
Input Rise & Fall Times (10% to 90%)	1.5 ns
Input Timing Reference Levels	(Figure 8a)
Output Reference Levels	0.35 VDD
Output Load, Timing Tests	0.35 VDD
	Figure 8b

CAPACITANCE [1,4]

SYMBOL	DESCRIPTION	TEST CONDITIONS	RATING
CIN	Input Capacitance	TA=25 deg C, F=1MHz, VDD=3.13 V	10 pF
COUT	Output Capacitance	TA=25 deg C, F=1MHz, VDD=3.13 V	10 pF

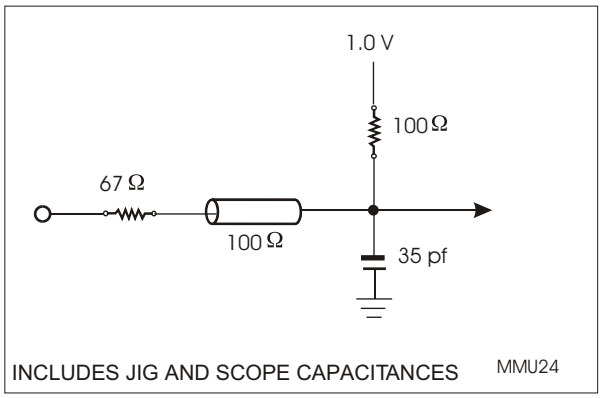


Figure 8a. Output Load Circuit

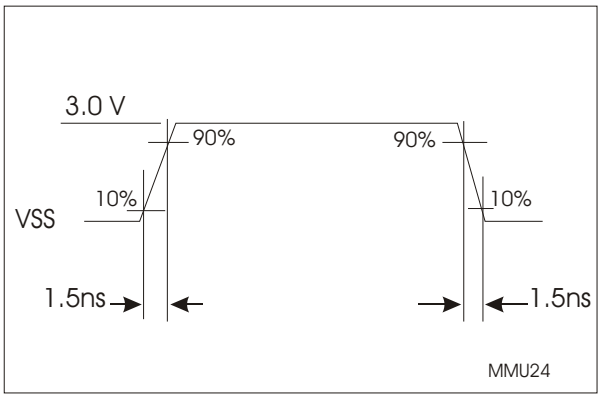


Figure 8b. Input Rise and Fall Times

NOTES:

1. All voltages are measured with respect to VSS.

2. Stresses greater than those listed under 'Absolute Maximum Ratings' may cause permanent damage to the device. This is a stress rating for transient conditions only. Functional operation of the device at these or any other conditions above those indicated in the ' Operating Range' of this specification is not implied.
Exposure to absolute maximum rating conditions for extended periods may affect reliability.

3. Outputs should not be shorted for more than 30 seconds. No more than one output should be shorted at any time.

4. Sample tested only

5. The MMU24 inputs are able to withstand a -1.0V undershoot for less than 10 ns per cycle.

6. An internal pull-up resistor is attached to the DB[7:0] pins.

7. An internal pull-down resistor is attached to the CCOMR, and CCOMI pins.

8. IDD is dependent upon actual output loading and cycle rates. Specific values are with outputs open.

AC ELECTRICAL CHARACTERISTICS

		80 MHz			100 MHz			
SIGNAL	DESCRIPTION	MIN	TYP	MAX	MIN	TYP	MAX	UNIT
t_{CYC}	SYSCLK Cycle Time	12.5			10			ns
t_{CLSY}	Clock Low Time (SYSCLK)	6.25			5			ns
t_{CHSY}	Clock High Time (SYSCLK)	6.25			5			ns
t_{RESET}	RESET High Time	30			25			ns
t_{SSTART}	START Setup Time (SYSCLK)	8			7			ns
t_{HSTART}	START Hold Time (SYSCLK)	0			0			ns
$t_{SSYSCLKEN}$	$\overline{\text{SYSCLKEN}}$ Setup Time (SYSCLK)	8			7			ns
$t_{HSYSCLKEN}$	$\overline{\text{SYSCLKEN}}$ Hold Time (SYSCLK)	0			0			ns
t_{OEADR}	ADROE Low to ADR[19:0] Low-Z			9			7	ns
t_{OEZADR}	ADROE High to ADR[19:0] High-Z			9			7	ns
t_{ODADR}	ADR[19:0] Output Valid Delay Time (SYSCLK)			9			7	ns
t_{OHADR}	ADR[19:0] Output Hold (SYSCLK)	3			3			ns
t_{ODMOE}	$\overline{\text{MEMOE}}$ Output Delay Time (SYSCLK)			9			7	ns
t_{ODMW}	$\overline{\text{MEMW}}$ Output Delay Time (SYSCLK)			9			7	ns
t_{ODTC}	TC, Output Delay Time (SYSCLK)			9			7	ns
t_{ODTCP}	TCP, Output Delay Time (SYSCLK			9			7	ns
t_{ODACT}	$\overline{\text{ACTIVE}}$, Output Delay Time (SYSCLK)			9			7	ns
t_{ODPO}	PO, Output Delay Time (SYSCLK)			10			7.5	ns
t_{HZPO}	PO, High Impedance, Output Delay Time (SYSCLK)*			10			7.5	ns
t_{LZPO}	PO, Low Impedance, Output Delay Time (SYSCLK)*			10			7.5	ns
$t_{ODCCOMR}$	CCOMR, Output Delay Time (SYSCLK)*			10			7.5	ns
$t_{HZCCOMR}$	CCOMR, High Impedance, Output Delay Time (SYSCLK)*			10			7.5	ns
$t_{LZCCOMR}$	CCOMR, Low Impedance, Output Delay Time (SYSCLK)*			10			7.5	ns
$t_{ODCCOMI}$	CCOMI, Output Delay Time (SYSCLK)*			10			7.5	ns
$t_{HZCCOMI}$	CCOMI, High Impedance, Output Delay Time (SYSCLK)*			10			7.5	ns
$t_{LZCCOMI}$	CCOMI, Low Impedance, Output Delay Time (SYSCLK)*			10			7.5	ns
t_{ODSWAP}	SWAP, Output Delay Time (SYSCLK)*			10			7.5	ns
t_{HZSWAP}	SWAP, High Impedance, Output Delay Time (SYSCLK)*			10			7.5	ns
t_{LZSWAP}	SWAP, Low Impedance, Output Delay Time (SYSCLK)*			10			7.5	ns
t_{LRW}	R/$\overline{\text{W}}$ Low Time	12.5			10			ns
t_{HRW}	R/$\overline{\text{W}}$ High Time	12.5			10			ns
t_{SCS}	$\overline{\text{CS}}$ Setup Time (R/$\overline{\text{W}}$)	24			18			ns
t_{HCS}	$\overline{\text{CS}}$ Hold Time (R/$\overline{\text{W}}$)	0			0			ns
t_{HZCSDB}	$\overline{\text{CS}}$ High to DB[7:0] High-Z*			15			12	ns
t_{LZCSDB}	$\overline{\text{CS}}$ Low to DB[7:0] Low-Z*			15			12	ns
t_{ODCSDB}	$\overline{\text{CS}}$ Low to DB[7:0] Output Valid			22			17	ns
$t_{HZDIRDB}$	DIR Low to DB[7:0] High-Z*			15			12	ns
$t_{LZDIRDB}$	DIR High to DB[7:0] Low-Z*			15			12	ns
$t_{ODDIRDB}$	DIR High to DB[7:0] Output Valid			22			17	ns
t_{SA1A0I}	A1A0 Setup Time (R/$\overline{\text{W}}$) Internal Address Mode	24			18			ns
t_{SA1A0X}	A1A0 Setup Time (R/$\overline{\text{W}}$) External Address Mode	24			18			ns
t_{HA1A0}	A1A0 Hold Time (R/$\overline{\text{W}}$)	0			0			ns
$t_{ODA1A0DB}$	A1A0 to DB[7:0] Output Valid			22			17	ns
t_{SDB}	DB[7:0] Setup Time (R/$\overline{\text{W}}$)	24			18			ns
t_{HDB}	DB[7:0] Hold Time (R/$\overline{\text{W}}$)	0	0		0	0		ns

Figure 10. Timing Diagram

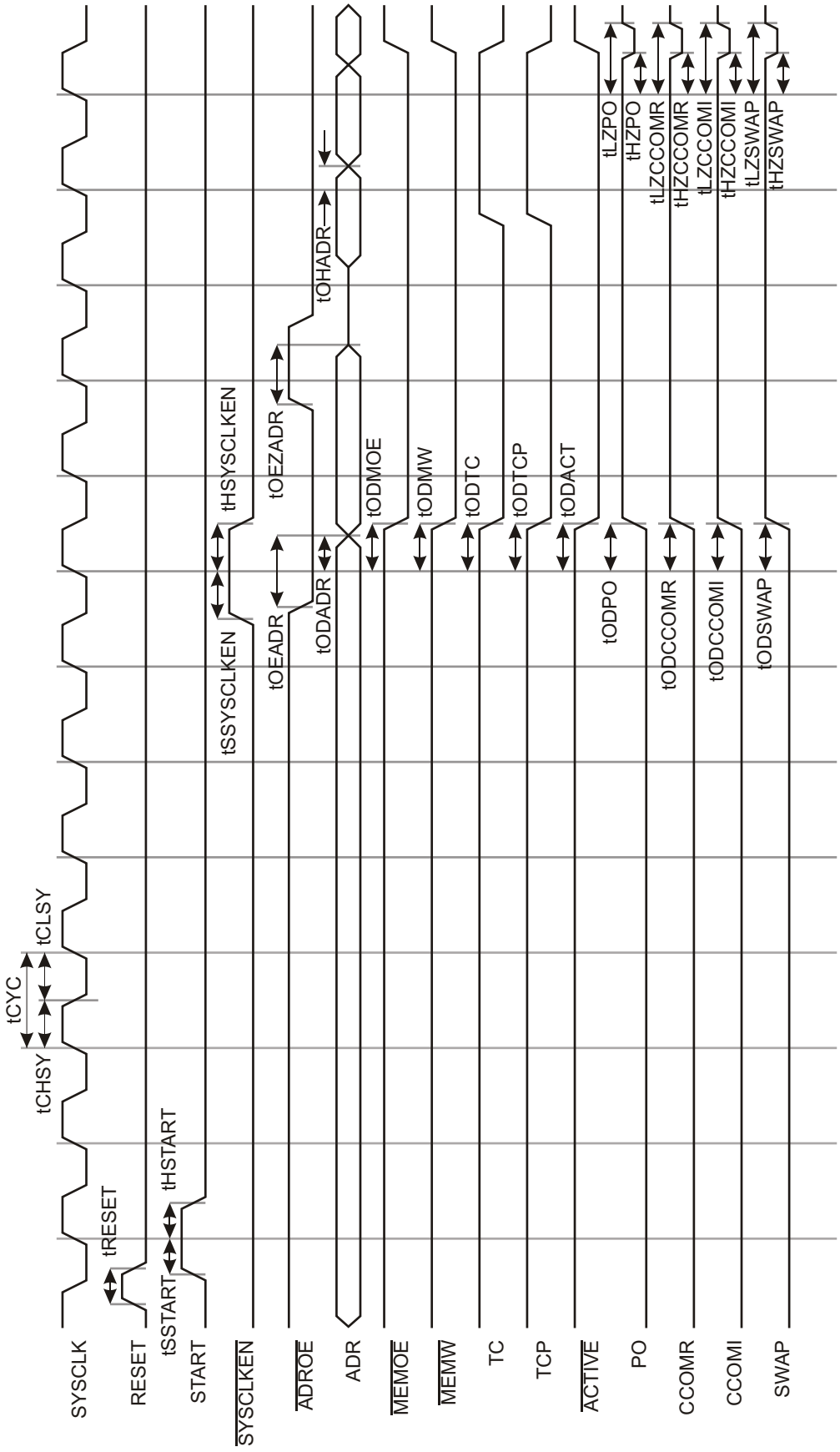
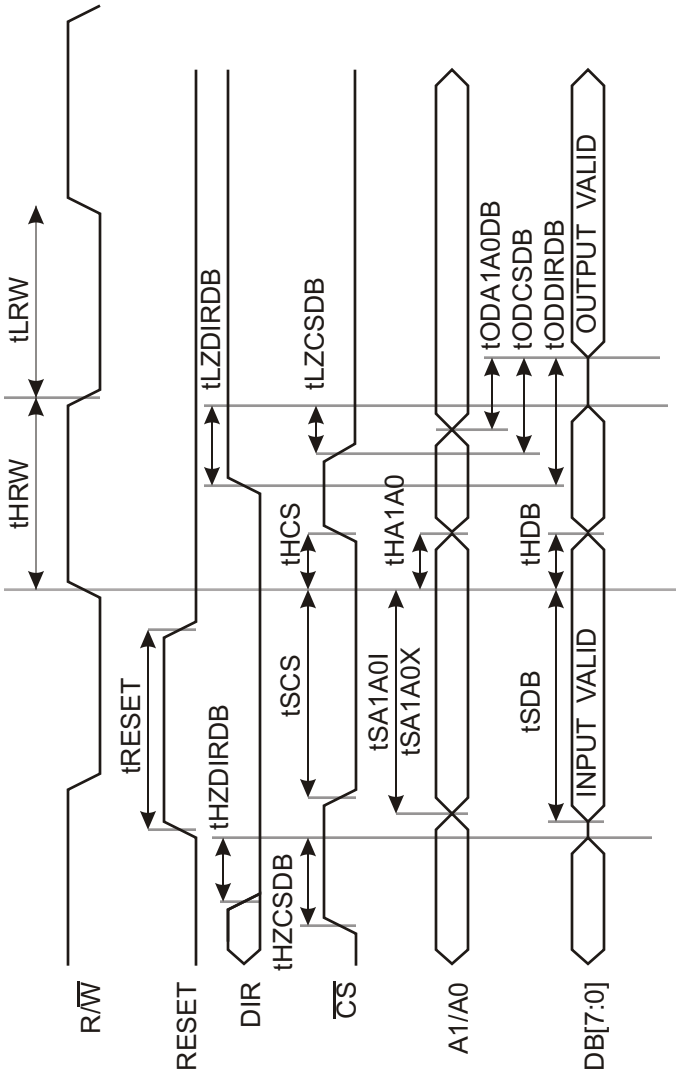
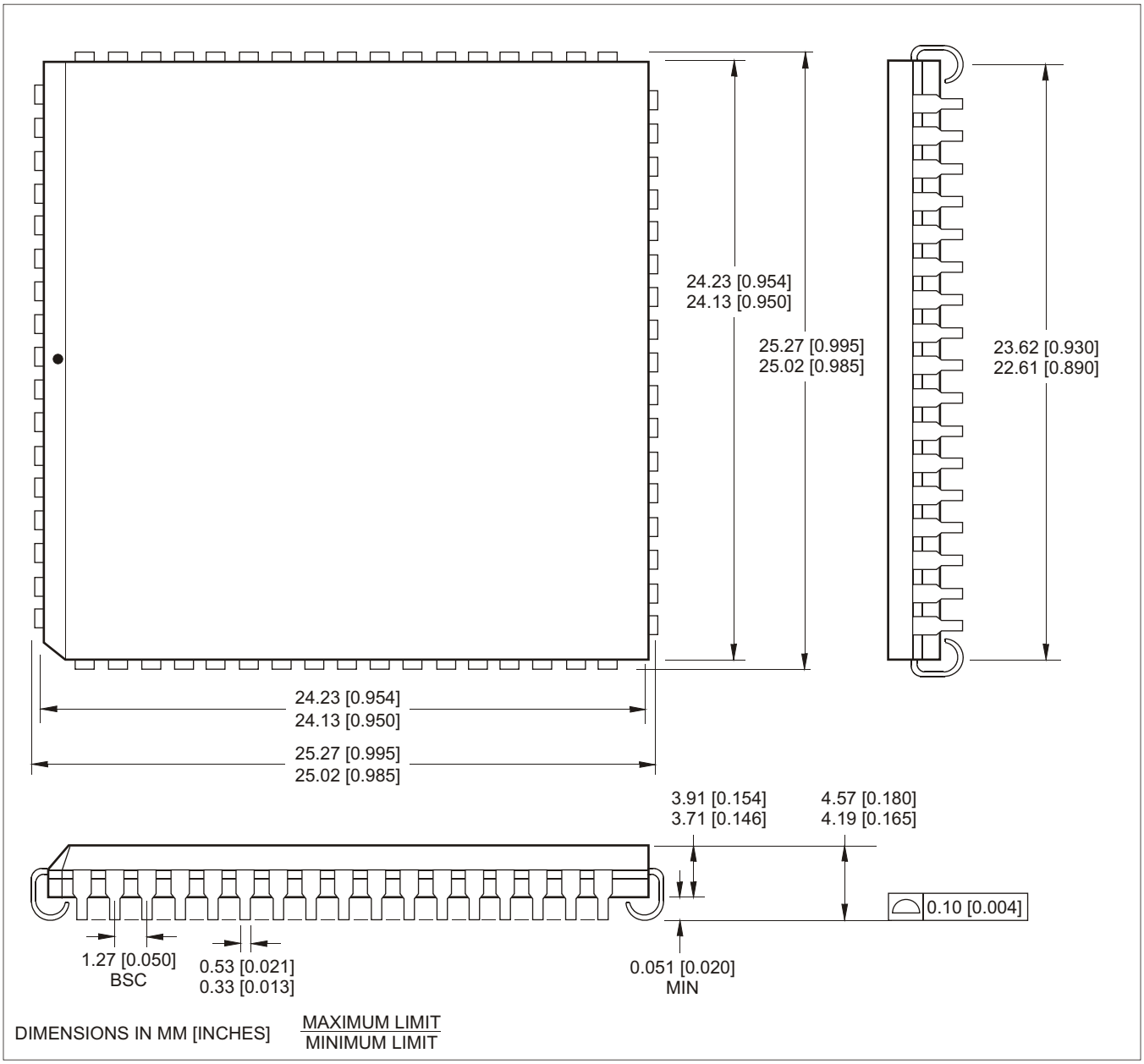


Figure 10. Timing Diagram (cont.)



PACKAGE DIAGRAM: 68-PIN, PLCC



MMU24 80-PIN TQFP PIN LIST

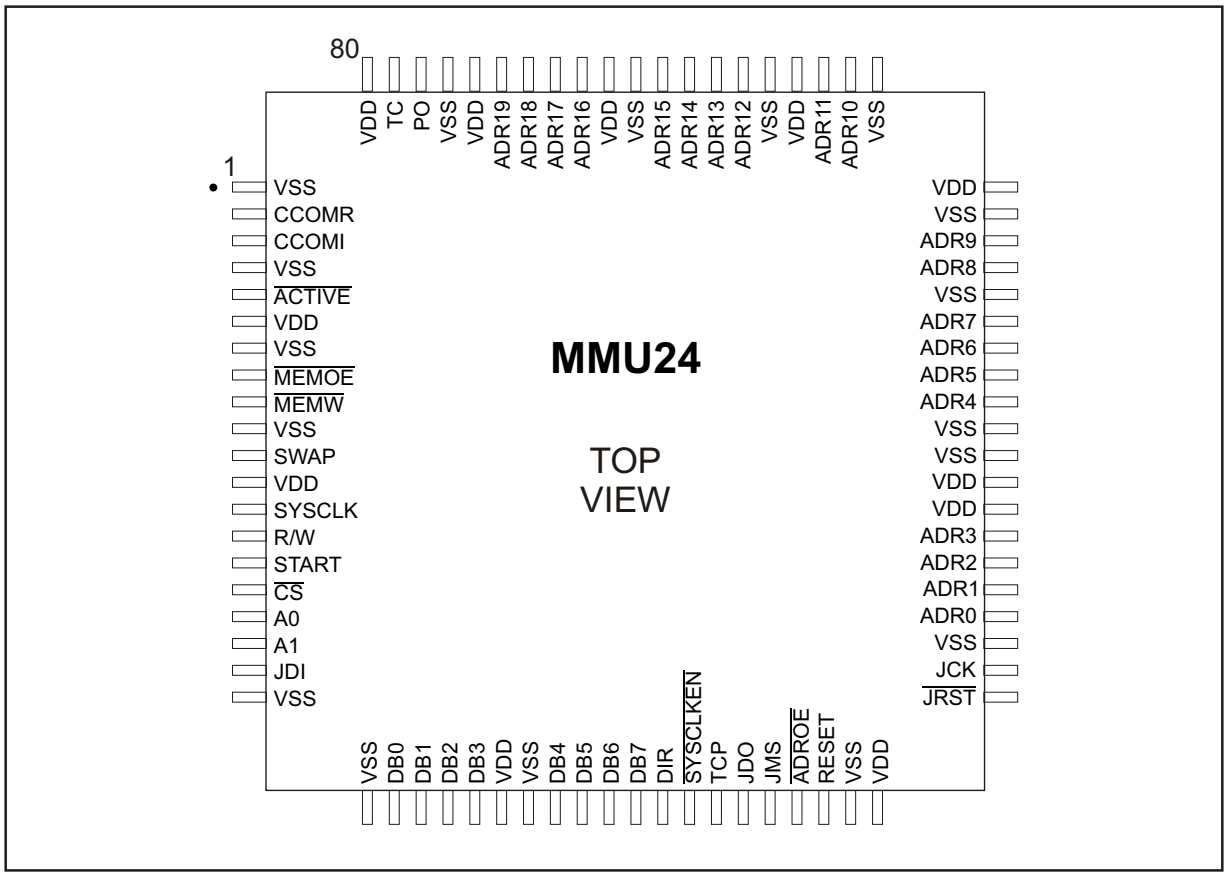
PIN	SIGNAL
1	VSS
2	CCOMR
3	CCOMI
4	VSS
5	$\overline{\text{ACTIVE}}$
6	VDD
7	VSS
8	$\overline{\text{MEMOE}}$
9	$\overline{\text{MEMW}}$
10	VSS
11	SWAP
12	VDD
13	SYSCLK
14	R/W
15	START
16	$\overline{\text{CS}}$
17	A0
18	A1
19	JDI
20	VSS

PIN	SIGNAL
21	VSS
22	DB0
23	DB1
24	DB2
25	DB3
26	VDD
27	VSS
28	DB4
29	DB5
30	DB6
31	DB7
32	DIR
33	$\overline{\text{SYSCLKEN}}$
34	TCP
35	JDO
36	JMS
37	$\overline{\text{ADROE}}$
38	RESET
39	VSS
40	VDD

PIN	SIGNAL
41	$\overline{\text{JRST}}$
42	JCK
43	VSS
44	ADR0
45	ADR1
46	ADR2
47	ADR3
48	VDD
49	VDD
50	VSS
51	VSS
52	ADR4
53	ADR5
54	ADR6
55	ADR7
56	VSS
57	ADR8
58	ADR9
59	VSS
60	VDD

PIN	SIGNAL
61	VSS
62	ADR10
63	ADR11
64	VDD
65	VSS
66	ADR12
67	ADR13
68	ADR14
69	ADR15
70	VSS
71	VDD
72	ADR16
73	ADR17
74	ADR18
75	ADR19
76	VDD
77	VSS
78	PO
79	TC
80	VDD

PACKAGE DRAWING: 80-PIN TQFP PIN CONNECTIONS



ORDERING INFORMATION

#	-	#	-	#	-	#
Device		Package		Speed		Temperature

- Temperature: M=Military, I=Industrial, C=Commercial
- Speed: 20, 40, 60, 80, 100 Operating Frequency (MHz)
- Package: Y = 68-lead PLCC T= 80-lead TQFP
- Device: MMU24

Example: MMU24-T-100-C (80-Lead TQFP, 100 MHz Operating Frequency, Commercial Temperature)

DSP Architectures
Transform Your World™

NORTH AMERICA

Digital Signal Processing Architectures Inc.
7902 NE St. Johns Rd., Bldg 102
Vancouver, WA 98665, USA

Phone: 360 573-4084
Fax: 360 573-4272
Web Page: http://www.DSParchitectures.com

