EC7010/INTRODUCTION TO EMBEDDED CONTROLLERS

UNIT II 16-BIT MICROCONTROLLER

DsPIC30F microcontroller- architecture, DSP engine, memory, parallel ports, system and power management, ADC, interrupt, PWM.

Textbooks

1. Zoran Milivojević, Djordje Šaponjić, “Programming dsPIC Microcontrollers in C”, MikroElectronica.

https://learn.mikroe.com/ebooks/dspicprogrammingc/

2. DsPIC30F microcontroller Reference Manual

www.sathieshkumar.com/tutorials

dsPIC30F Architecture

- 16-bit modified Harvard architecture processors with an enhanced instruction set, including significant support for DSP.
- Instructions are 24-bit wide, with a variable length opcode field.
- The program counter (PC) is 23-bits wide and addresses up to 4M x 24 bits of user program memory space.
- Single cycle instructions except program flow transfer instructions, double word and table instructions.
- The working registers are 16-bit wide and sixteen in number.
- The 16th working register (W15) operates as a software stack pointer.
- Each of the working registers can act as a data, address, or address offset register.

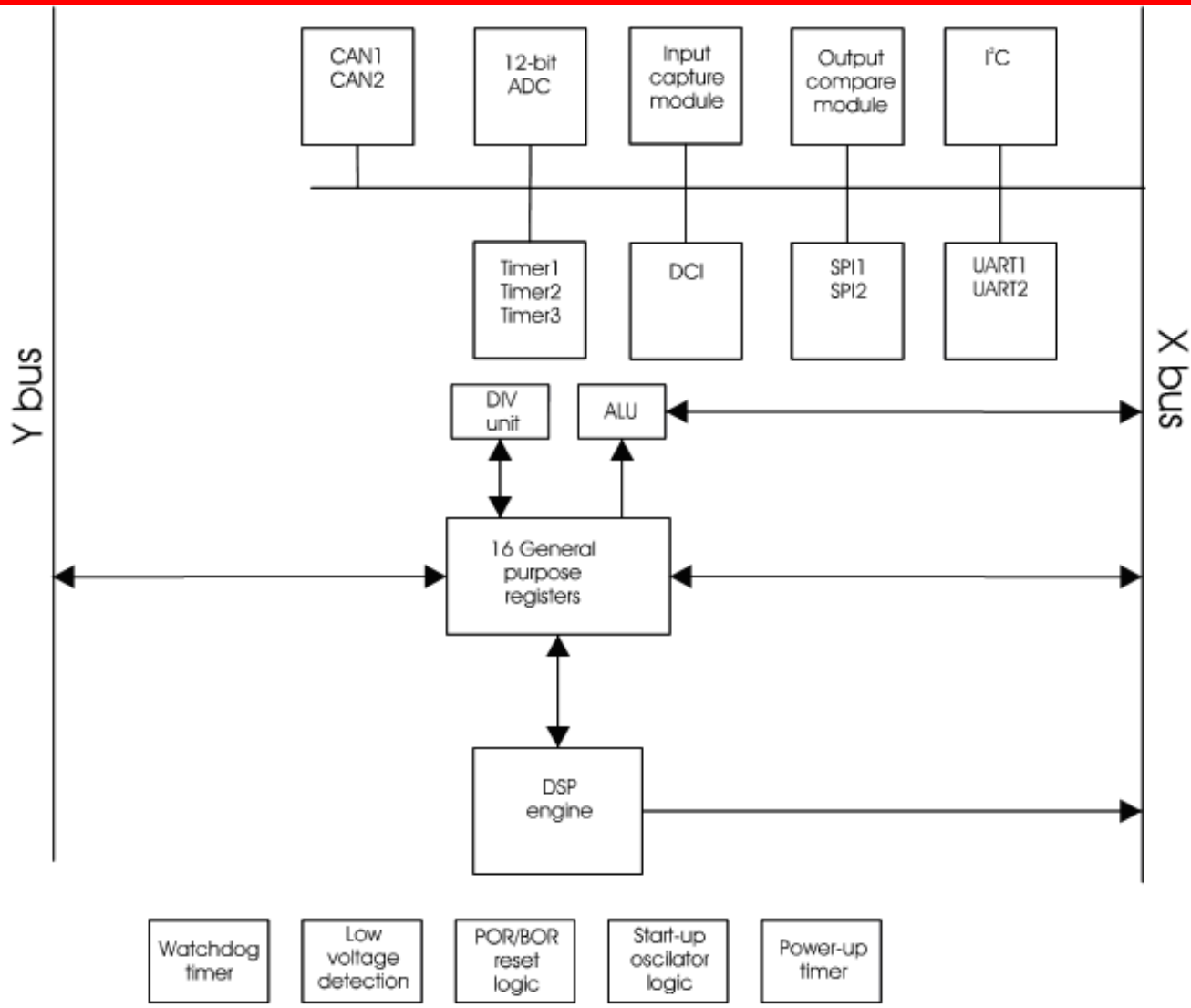
dsPIC30F Architecture

- The dsPIC30F instruction set has two classes of instructions: the MCU class of instructions and the DSP class of instructions.
- The data space can be addressed as 32K words or 64 Kbytes and is split into two blocks, referred to as X and Y data memory.
- Each memory block has its own independent Address Generation Unit (AGU).
- The MCU class of instructions operate solely through the X memory AGU, which accesses the entire memory map as one linear data space.
- Certain DSP instructions operate through the X and Y AGUs to support dual operand reads, which splits the data address space into two parts.
- The CPU supports Inherent (no operand), Relative, Literal, Memory Direct, Register Direct and Register Indirect Addressing modes.

dsPIC30F Architecture

- 3 operand instructions are supported.
- The DSP engine features a high speed, 17-bit by 17-bit multiplier, a 40-bit ALU, two 40-bit saturating accumulators and a 40-bit bi-directional barrel shifter.
- The barrel shifter is capable of shifting a 40-bit value up to 15 bits right, or up to 16 bits left, in a single cycle.
- The DSP instructions operate seamlessly with all other instructions and have been designed for optimal real-time performance.
- The dsPIC30F has a vectored exception scheme with up to 8 sources of non-maskable traps and 54 interrupt sources.
- Each interrupt source can be assigned to one of seven priority levels.

dsPIC30F Architecture

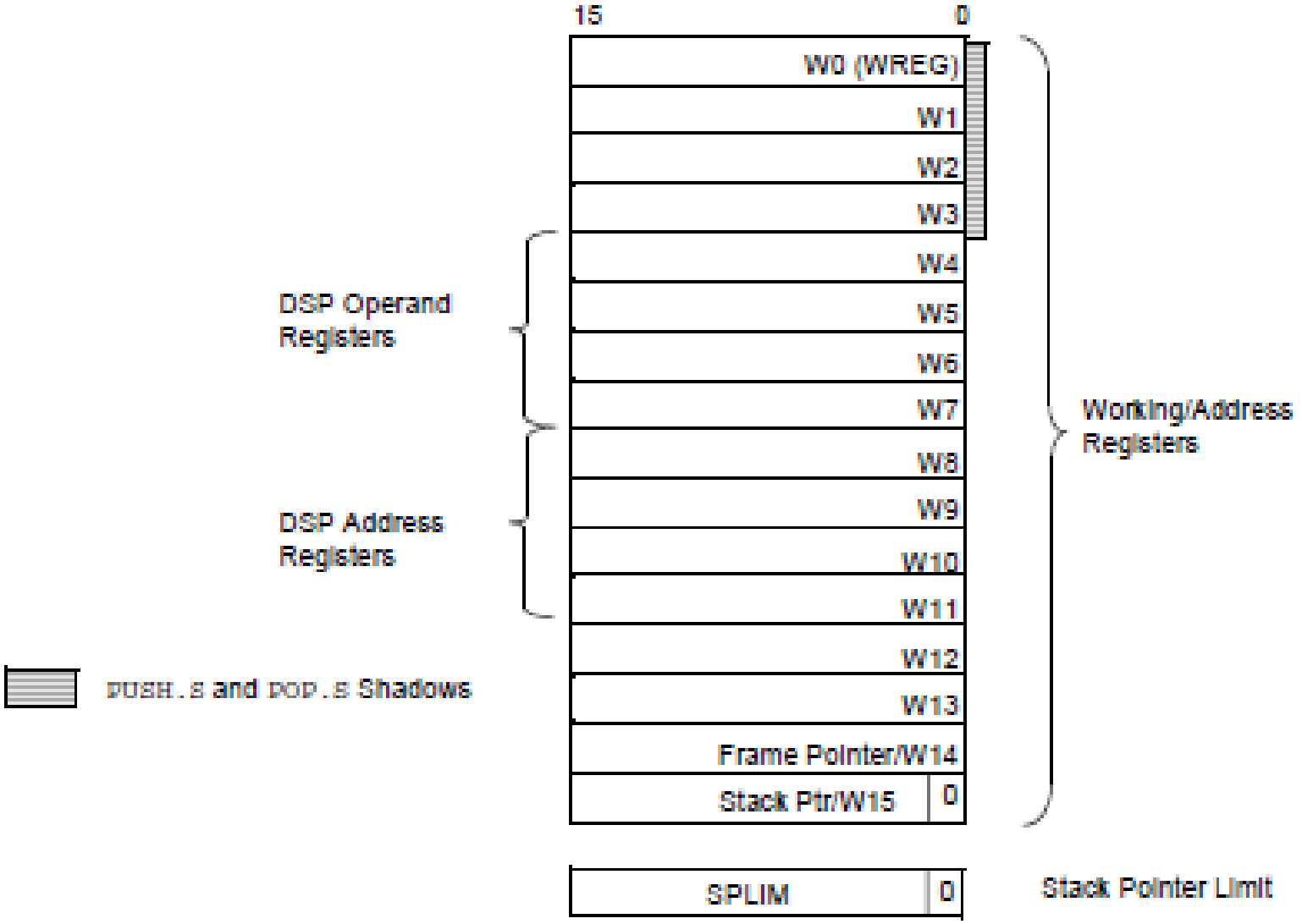


DCI- Data Convertor Interface: Allows simple interfacing of devices such as audio coder/decoders (codecs), A/D convertors and D/A convertors.

dsPIC30F Architecture: Programmer Model

Register(s) Name	Description
W0 through W15	Working register array
ACCA, ACCB	40-bit DSP Accumulators
PC	23-bit Program Counter
SR	ALU and DSP Engine Status register
SPLIM	Stack Pointer Limit Value register
TBLPAG	Table Memory Page Address register
PSVPAG	Program Space Visibility Page Address register
RCOUNT	REPEAT Loop Count register
DCOUNT	DO Loop Count register
DOSTART	DO Loop Start Address register
DOEND	DO Loop End Address register
CORCON	Contains DSP Engine and DO Loop control bits

dsPIC30F Architecture: Programmer Model



dsPIC30F Architecture: Programmer Model

- The 16 working (W) registers can function as data, address or address offset registers.
- The function of a W register is determined by the Addressing mode of the instruction that accesses it.
- The dsPIC30F instruction set can be divided into two instruction types: register and file register instructions.
- Register instructions can use each W register as a data value or an address offset value.
- W0 is a special working register because it is the only working register that can be used in file register instructions.
- File register instructions operate on a specific memory address contained in the instruction opcode and W0.

dsPIC30F Architecture: Programmer Model

- W1-W15 cannot be specified as a target register in file register instructions.
- The PUSH.S and POP.S instructions are useful for fast context save/restore during a function call or Interrupt Service Routine (ISR).
- The PUSH.S instruction will transfer the following register values into their respective shadow registers:

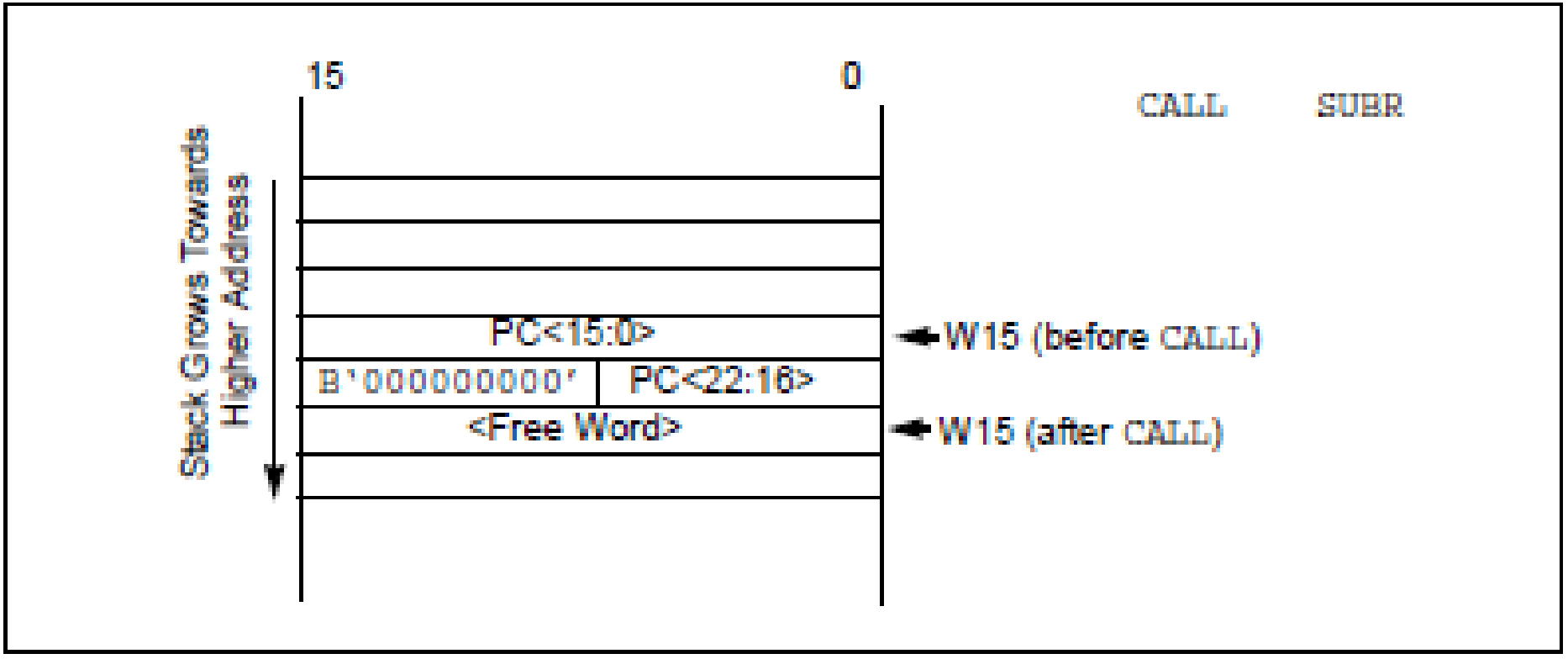
1. W0...W3
2. SR (N, OV, Z , C, DC bits only)

dsPIC30F Architecture: Programmer Model: Software Stack Pointer

- W15 serves as a dedicated software stack pointer and is automatically modified by exception processing, subroutine calls and returns.
- When the PC is pushed onto the stack, PC<15:0> is pushed onto the first available stack word, then PC<22:16> is pushed into the second available stack location.
- For a PC push during any CALL instruction, the MSByte of the PC is zero-extended before the push as shown in Figure 2-3.

dsPIC30F Architecture: Programmer Model

Figure 2-3: Stack Operation for a CALL Instruction



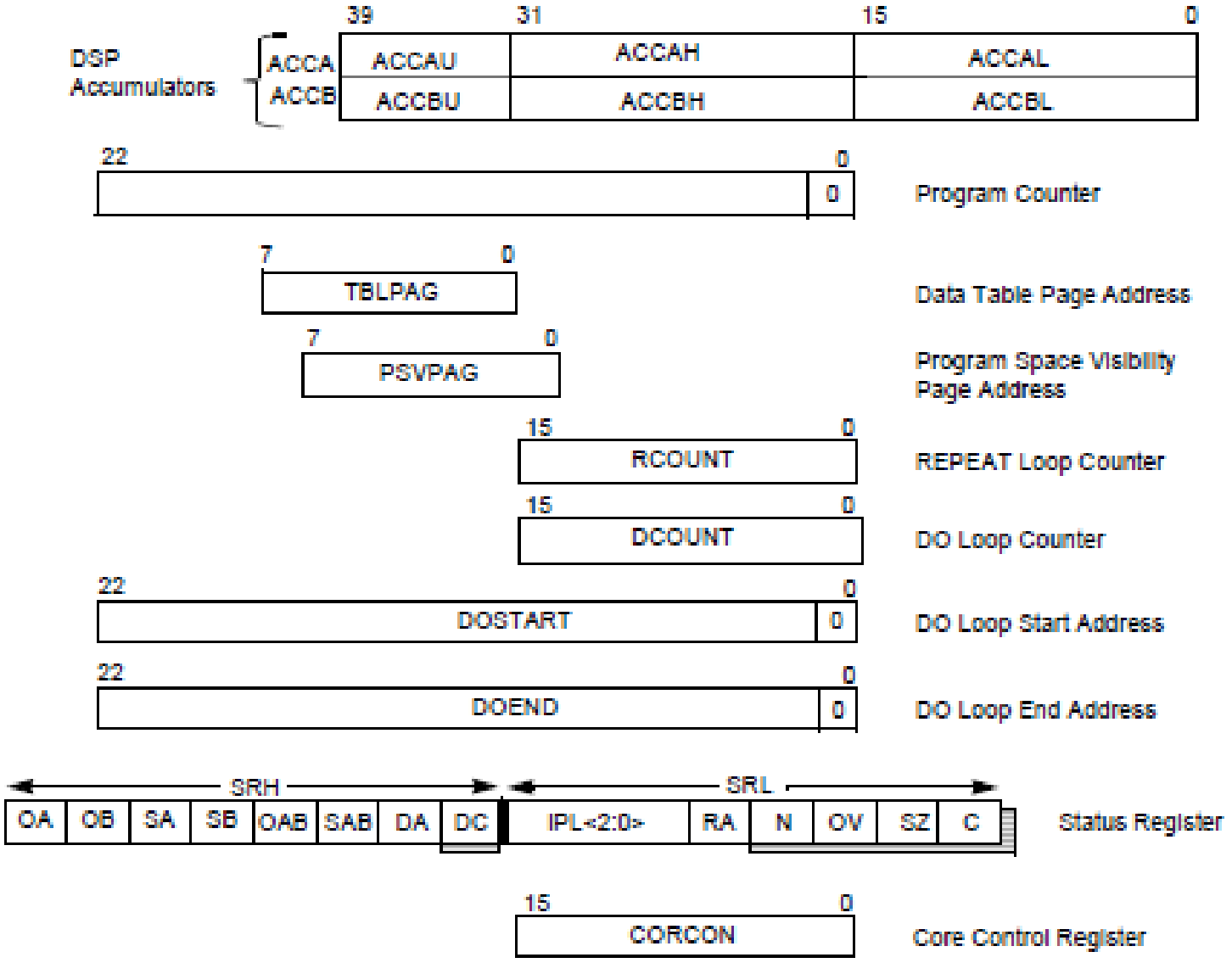
dsPIC30F Architecture: Programmer Model- W14 Software Stack Frame Pointer

- A frame is a user defined section of memory in the stack that is used by a single subroutine.
- W14 is a special working register because it can be used as a stack frame pointer with the LNK (link) and ULNK (unlink) instructions.
- W14 can be used in a normal working register by instructions when it is not used as a frame pointer.

dsPIC30F Architecture: Programmer Model- Stack Pointer Overflow

- There is a stack limit register (SPLIM) associated with the stack pointer that is reset to 0x0000.
- SPLIM is a 16-bit register, but SPLIM<0> is fixed to '0' because all stack operations must be word aligned.
- The stack overflow check will not be enabled until a word write to SPLIM occurs, after which time it can only be disabled by a device Reset.

dsPIC30F Architecture: Programmer Model



dsPIC30F Architecture: Programmer Model- Status Register

- The dsPIC30F CPU has a 16-bit status register (SR), the LSByte of which is referred to as the lower status register (SRL).
- The upper byte of SR is referred to as SRH.
- SRL contains all the MCU ALU operation status flags, plus the CPU interrupt priority status bits, IPL<2:0> and the REPEAT loop active status bit, RA (SR<4>).
- SRH contains the DSP Adder/Subtractor status bits, the DO loop active bit, DA (SR<9>) and the Digit Carry bit, DC (SR<8>).

dsPIC30F Architecture: Programmer Model- Status Register

➢The SR bits are readable/writable with the following exceptions:

1. The DA bit (SR<8>): DA is a read only bit.

2. The RA bit (SR<4>): RA is a read only bit.

3. The OA, OB (SR<15:14>) and OAB (SR<11>) bits: These bits are read only and can only be modified by the DSP engine hardware.

4. The SA, SB (SR<13:12>) and SAB (SR<10>) bits: These are read and clear only and can only be set by the DSP engine hardware.

dsPIC30F Architecture: Programmer Model- Status Register

Register 2-1: SR: CPU Status Register

Upper Byte:							
R-0	R-0	R/C-0	R/C-0	R-0	R/C-0	R -0	R/W-0
OA	OB	SA	SB	OAB	SAB	DA	DC
bit 15							bit 8

Lower Byte: (SRL)							
R/W-0(2)	R/W-0(2)	R/W-0(2)	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL<2:0>			RA	N	OV	Z	C
bit 7							bit 0

bit 15 **OA**: Accumulator A Overflow Status bit
1 = Accumulator A overflowed
0 = Accumulator A has not overflowed

bit 14 **OB**: Accumulator B Overflow Status bit
1 = Accumulator B overflowed
0 = Accumulator B has not overflowed

bit 13 **SA**: Accumulator A Saturation 'Sticky' Status bit
1 = Accumulator A is saturated or has been saturated at some time
0 = Accumulator A is not saturated

Note: This bit may be read or cleared (not set).

dsPIC30F Architecture: Programmer Model- Status Register

bit 12 **SB**: Accumulator B Saturation 'Sticky' Status bit
1 = Accumulator B is saturated or has been saturated at some time
0 = Accumulator B is not saturated

Note: This bit may be read or cleared (not set).

bit 11 **OAB**: OA || OB Combined Accumulator Overflow Status bit
1 = Accumulators A or B have overflowed
0 = Neither Accumulators A or B have overflowed

bit 10 **SAB**: SA || SB Combined Accumulator 'Sticky' Status bit
1 = Accumulators A or B are saturated or have been saturated at some time in the past
0 = Neither Accumulator A or B are saturated

Note: This bit may be read or cleared (not set). Clearing this bit will clear SA and SB.

bit 9 **DA**: DO Loop Active bit
1 = DO loop in progress
0 = DO loop not in progress

bit 8 **DC**: MCU ALU Half Carry/$\overline{\text{Borrow}}$ bit
1 = A carry-out from the 4th low order bit (for byte-sized data) or 8th low order bit (for word-sized data) of the result occurred
0 = No carry-out from the 4th low order bit (for byte-sized data) or 8th low order bit (for word-sized data) of the result occurred

dsPIC30F Architecture: Programmer Model- Status Register

bit 7-5 **IPL<2:0>**: CPU Interrupt Priority Level Status bits(1)

111 = CPU Interrupt Priority Level is 7 (15). User interrupts disabled.
110 = CPU Interrupt Priority Level is 6 (14)
101 = CPU Interrupt Priority Level is 5 (13)
100 = CPU Interrupt Priority Level is 4 (12)
011 = CPU Interrupt Priority Level is 3 (11)
010 = CPU Interrupt Priority Level is 2 (10)
001 = CPU Interrupt Priority Level is 1 (9)
000 = CPU Interrupt Priority Level is 0 (8)

Note 1: The IPL<2:0> bits are concatenated with the IPL<3> bit (CORCON<3>) to form the CPU Interrupt Priority Level. The value in parentheses indicates the IPL if IPL<3> = 1. User interrupts are disabled when IPL<3> = 1.

2: The IPL<2:0> status bits are read only when NSTDIS = 1 (INTCON1<15>).

bit 4 **RA**: REPEAT Loop Active bit

1 = REPEAT loop in progress
0 = REPEAT loop not in progress

dsPIC30F Architecture: Programmer Model- Status Register

bit 3 **N**: MCU ALU Negative bit
1 = Result was negative
0 = Result was non-negative (zero or positive)

bit 2 **OV**: MCU ALU Overflow bit
This bit is used for signed arithmetic (2's complement). It indicates an overflow of the magnitude which causes the sign bit to change state.
1 = Overflow occurred for signed arithmetic (in this arithmetic operation)
0 = No overflow occurred

bit 1 **Z**: MCU ALU Zero bit
1 = An operation which effects the Z bit has set it at some time in the past
0 = The most recent operation which effects the Z bit has cleared it (i.e., a non-zero result)

bit 0 **C**: MCU ALU Carry/$\overline{\text{Borrow}}$ bit
1 = A carry-out from the Most Significant bit of the result occurred
0 = No carry-out from the Most Significant bit of the result occurred

dsPIC30F Architecture: Programmer Model- Core Control Register

- The CORCON register contains bits that control the operation of the DSP multiplier and DO loop hardware.
- The CORCON register also contains the IPL3 status bit, which is concatenated with IPL<2:0> (SR<7:5>), to form the CPU Interrupt Priority Level.

Register 2-2: CORCON: Core Control Register

Upper Byte:

U-0	U-0	U-0	R/W-0	R/W-0	R-0	R-0	R-0
—	—	—	US	EDT	DL<2:0>		
bit 15							bit 8

Lower Byte:

R/W-0	R/W-0	R/W-1	R/W-0	R/C-0	R/W-0	R/W-0	R/W-0
SATA	SATB	SATDW	ACCSAT	IPL3	PSV	RND	IF
bit 7							bit 0

dsPIC30F Architecture: Programmer Model- Core Control Register

bit 15-13 **Unimplemented**: Read as '0'

bit 12 **US**: DSP Multiply Unsigned/Signed Control bit
1 = DSP engine multiplies are unsigned
0 = DSP engine multiplies are signed

bit 11 **EDT**: Early DO Loop Termination Control bit
1 = Terminate executing DO loop at end of current loop iteration
0 = No effect

Note: This bit will always read as '0'.

bit 10-8 **DL<2:0>**: DO Loop Nesting Level Status bits
111 = 7 DO loops active
.
.
001 = 1 DO loop active
000 = 0 DO loops active

dsPIC30F Architecture: Programmer Model- Core Control Register

bit 7 **SATA**: AccA Saturation Enable bit
1 = Accumulator A saturation enabled
0 = Accumulator A saturation disabled

bit 6 **SATB**: AccB Saturation Enable bit
1 = Accumulator B saturation enabled
0 = Accumulator B saturation disabled

bit 5 **SATDW**: Data Space Write from DSP Engine Saturation Enable bit
1 = Data space write saturation enabled
0 = Data space write saturation disabled

bit 4 **ACCSAT**: Accumulator Saturation Mode Select bit
1 = 9.31 saturation (super saturation)
0 = 1.31 saturation (normal saturation)

bit 3 **IPL3**: CPU Interrupt Priority Level Status bit 3
1 = CPU interrupt priority level is greater than 7
0 = CPU interrupt priority level is 7 or less

Note: The IPL3 bit is concatenated with the IPL<2:0> bits (SR<7:5>) to form the CPU interrupt priority level.

dsPIC30F Architecture: Programmer Model- Core Control Register

bit 2 **PSV**: Program Space Visibility in Data Space Enable bit
1 = Program space visible in data space
0 = Program space not visible in data space

bit 1 **RND**: Rounding Mode Select bit
1 = Biased (conventional) rounding enabled
0 = Unbiased (convergent) rounding enabled

bit 0 **IF**: Integer or Fractional Multiplier Mode Select bit
1 = Integer mode enabled for DSP multiply ops
0 = Fractional mode enabled for DSP multiply ops

dsPIC30F Architecture: Programmer Model- TBLPAG: Table Page Register

- The TBLPAG register is used to hold the upper 8 bits of a program memory address during table read and write operations.
- Table instructions are used to transfer data between program memory space and data memory space.

dsPIC30F Architecture: Programmer Model- PSVPAG: Program Space Visibility Page Register

- Program space visibility allows the user to map a 32-Kbyte section of the program memory space into the upper 32 Kbytes of data address space.
- This feature allows transparent access of constant data through dsPIC30F instructions that operate on data memory.

DSP Engine

- The DSP engine is a block of hardware which is fed data from the W register array but contains its own specialized result registers.
- The DSP engine is driven from the same instruction decoder that directs the MCU ALU.
- The DSP engine consists of the following components:
  - high speed 17-bit x 17-bit multiplier
  - barrel shifter
  - 40-bit adder/subtractor
  - two target accumulator registers
  - rounding logic with Selectable modes
  - saturation logic with Selectable modes

DSP Engine

Data input to the DSP engine is derived from one of the following sources:

1. Directly from the W array (registers W4, W5, W6 or W7) for dual source operand DSP instructions. Data values for the W4, W5, W6 and W7 registers are pre-fetched via the X and Y memory data buses.

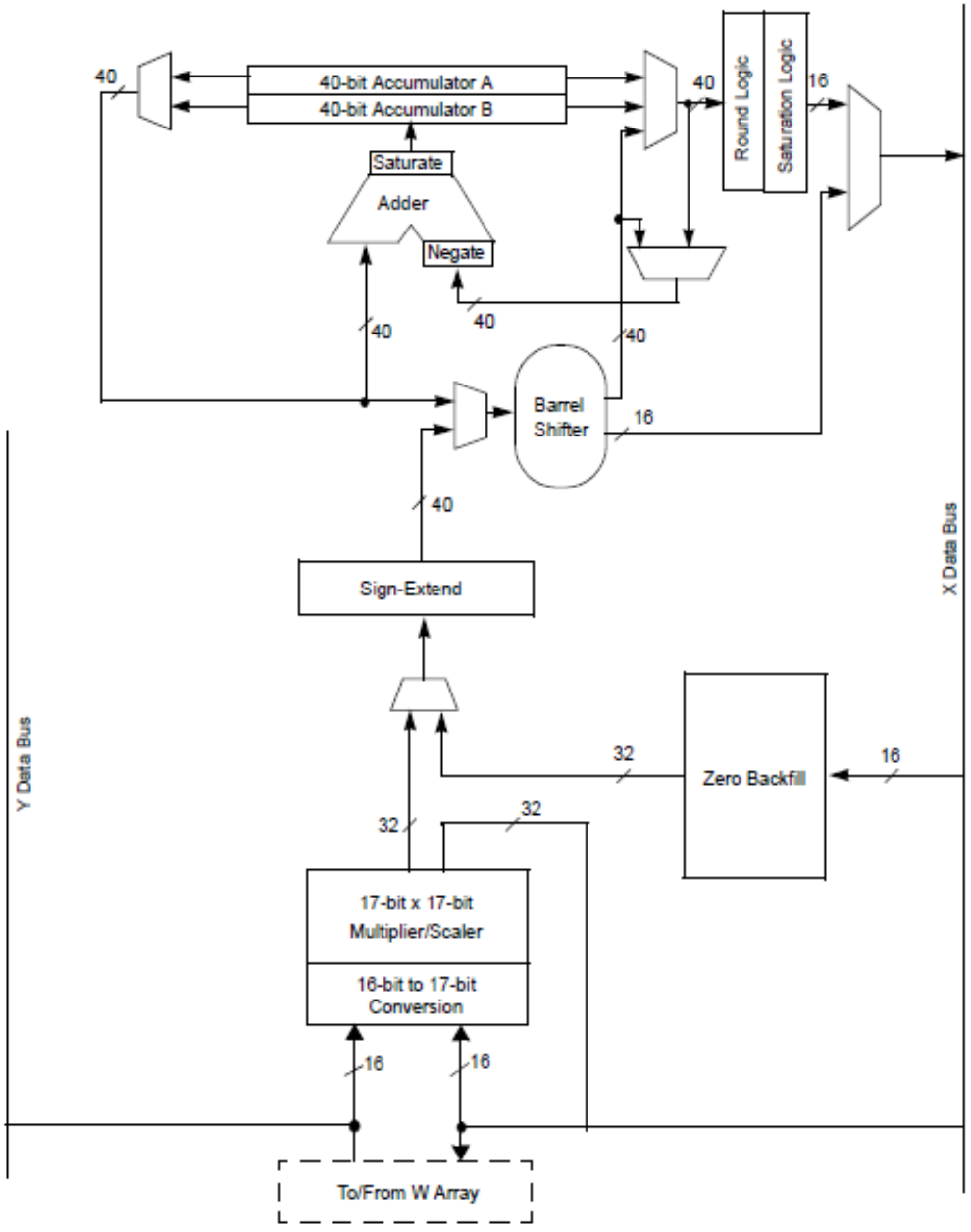
2. From the X memory data bus for all other DSP instructions.

Data output from the DSP engine is written to one of the following destinations:

1. The target accumulator, as defined by the DSP instruction being executed.

2. The X memory data bus to any location in the data memory address space.

DSP Engine


40
40-bit Accumulator A
40-bit Accumulator B
40
Round Logic
Saturation Logic
16
Saturate
Adder
Negate
40
40
40
Barrel Shifter
16
40
Sign-Extend
X Data Bus
Y Data Bus
32
Zero Backfill
16
32
32
17-bit x 17-bit Multiplier/Scaler
16-bit to 17-bit Conversion
16
16
To/From W Array

DSP Engine: Data Accumulators

➢There are two 40-bit data accumulators, ACCA and ACCB, that are the result registers for the DSP instructions.

➢ Each accumulator is memory mapped to three registers,

(where ‘x’ denotes the particular accumulator)

- ACCxL: ACCx<15:0>
- ACCxH: ACCx<31:16>
- ACCxU: ACCx<39:32>

DSP Engine: Multiplier

- The dsPIC30F features a 17-bit x 17-bit multiplier which is shared by both the MCU ALU and the DSP engine.
- The multiplier is capable of signed or unsigned operation.
- The multiplier takes in 16-bit input data and converts the data to 17-bits.
- Signed operands to the multiplier are sign-extended.
- Unsigned input operands are zero-extended.
- The 17-bit conversion logic is transparent to the user and allows the multiplier to support mixed sign and unsigned/signed multiplication.

DSP Engine: Multiplier

- The IF control bit (CORCON<0>) determines integer/fractional operation for the instructions.
- The IF bit does not affect MCU multiply instructions, which are always integer operations.
- The multiplier defaults to Fractional mode for DSP operations at a device Reset.

DSP Engine: Multiplier

➢The representation of data in hardware for each of these modes is as follows:

• Integer data is inherently represented as a signed two's complement value, where the MSbit is defined as a sign bit. Generally speaking, the range of an N-bit two's complement integer is -2^{N-1} to $2^{N-1} – 1$.

• Fractional data is represented as a two's complement fraction where the MSbit is defined as a sign bit and the radix point is implied to lie just after the sign bit. The range of an N-bit two's complement fraction with this implied radix point is -1.0 to $(1 – 2^{1-N})$.

DSP Engine: Multiplier

Figure 2-9: Integer and Fractional Representation of 0x4001

Different Representations of 0x4001

Integer:

0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
-2^{15}	2^{14}	2^{13}	2^{12}	. . . .											2^{0}

$0x4001 = 2^{14} + 2^{0} = 16385$

1.15 Fractional:

0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
-2^{0}	.	2^{-1}	2^{-2}	2^{-3}	. . .										2^{-15}

Implied Radix Point

$0x4001 = 2^{-1} + 2^{-15} = 0.500030518$

DSP Engine: Multiplier

Figure 2-10: Integer and Fractional Representation of 0xC002

Different Representations of 0xC002

Integer:

1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
-2^{15}	2^{14}	2^{13}	2^{12}	....											2^{0}

$0xC002 = -2^{15} + 2^{14} + 2^{0} = -32768 + 16384 + 2 = -16382$

1.15 Fractional:

1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
-2^{0}	.	2^{-1}	2^{-2}	2^{-3}	...										2^{-15}

Implied Radix Point

$0xC002 = -2^{0} + 2^{-1} + 2^{-14} = -1 + 0.5 + 0.000061035 = -0.499938965$

DSP Engine: Multiplier

Table 2-2: dsPIC30F Data Ranges

Register Size	Integer Range	Fraction Range
16-bit	-32768 to 32767	-1.0 to $(1.0 - 2^{-15})$ (Q.15 Format)
32-bit	-2,147,483,648 to 2,147,483,647	-1.0 to $(1.0 - 2^{-31})$ (Q.31 Format)
40-bit	-549,755,813,888 to 549,755,813,887	-256.0 to $(256.0 - 2^{-31})$ (Q.31 Format with 8 Guard bits)

DSP Engine: DSP Multiply Instructions

Table 2-3: DSP Instructions that Utilize the Multiplier

DSP Instruction	Description	Algebraic Equivalent
MAC	Multiply and Add to Accumulator OR Square and Add to Accumulator	$a = a + b*c$ $a = a + b^2$
MSC	Multiply and Subtract from Accumulator	$a = a - b*c$
MPY	Multiply	$a = b*c$
MPY.N	Multiply and Negate Result	$a = -b*c$
ED	Partial Euclidean Distance	$a = (b - c)^2$
EDAC	Add Partial Euclidean Distance to the Accumulator	$a = a + (b - c)^2$

Note: DSP instructions using the multiplier can operate in Fractional (1.15) or Integer modes.

DSP Engine: MCU Multiply Instructions

Table 2-4: MCU Instructions that Utilize the Multiplier

MCU Instruction	Description
MUL/MUL.UU	Multiply two unsigned integers
MUL.SS	Multiply two signed integers
MUL.SU/MUL.US	Multiply a signed integer with an unsigned integer

Note 1: MCU instructions using the multiplier operate only in Integer mode.

2: Result of an MCU multiply is 32-bits long and is stored in a pair of W registers.

DSP Engine: Data Accumulator Adder/Subtractor

- The data accumulators have a 40-bit adder/subtractor with automatic sign extension logic for the multiplier result (if signed).
- It can select one of two accumulators (A or B) as its pre-accumulation source and post-accumulation destination.
- The data to be accumulated or loaded can optionally be scaled via the barrel shifter prior to accumulation.
- The 40-bit adder/subtractor may optionally negate one of its operand inputs to change the sign of the result (without changing the operands).
- The 40-bit adder/subtractor has an additional saturation block which controls accumulator data saturation, if enabled.

DSP Engine: Accumulator Status Bits

Table 2-5: Accumulator Overflow and Saturation Status Bits

Status Bit	Location	Description
OA	SR<15>	Accumulator A overflowed into guard bits (ACCA<39:32>)
OB	SR<14>	Accumulator B overflowed into guard bits(ACCB<39:32>)
SA	SR<13>	ACCA saturated (bit 31 overflow and saturation) *or* ACCA overflowed into guard bits and saturated (bit 39 overflow and saturation)
SB	SR<12>	ACCB saturated (bit 31 overflow and saturation) *or* ACCB overflowed into guard bits and saturated (bit 39 overflow and saturation)
OAB	SR<11>	OA logically ORed with OB
SAB	SR<10>	SA logically ORed with SB. *Clearing SAB will also clear SA and SB.*

DSP Engine: Saturation and Overflow Modes

➢The device supports three Saturation and Overflow modes.

1. **Accumulator 39-bit Saturation:** In this mode, the saturation logic loads the maximally positive 9.31 value (0x7FFFFFFFFF), or maximally negative 9.31 value (0x8000000000), into the target accumulator. The SA or SB bit is set and remains set until cleared by the user.

2. **Accumulator 31-bit Saturation:** In this mode, the saturation logic loads the maximally positive 1.31 value (0x007FFFFFFF) or maximally negative 1.31 value (0xFF80000000) into the target accumulator. The SA or SB bit is set and remains set until cleared by the user.

DSP Engine: Saturation and Overflow Modes

➢The device supports three Saturation and Overflow modes.

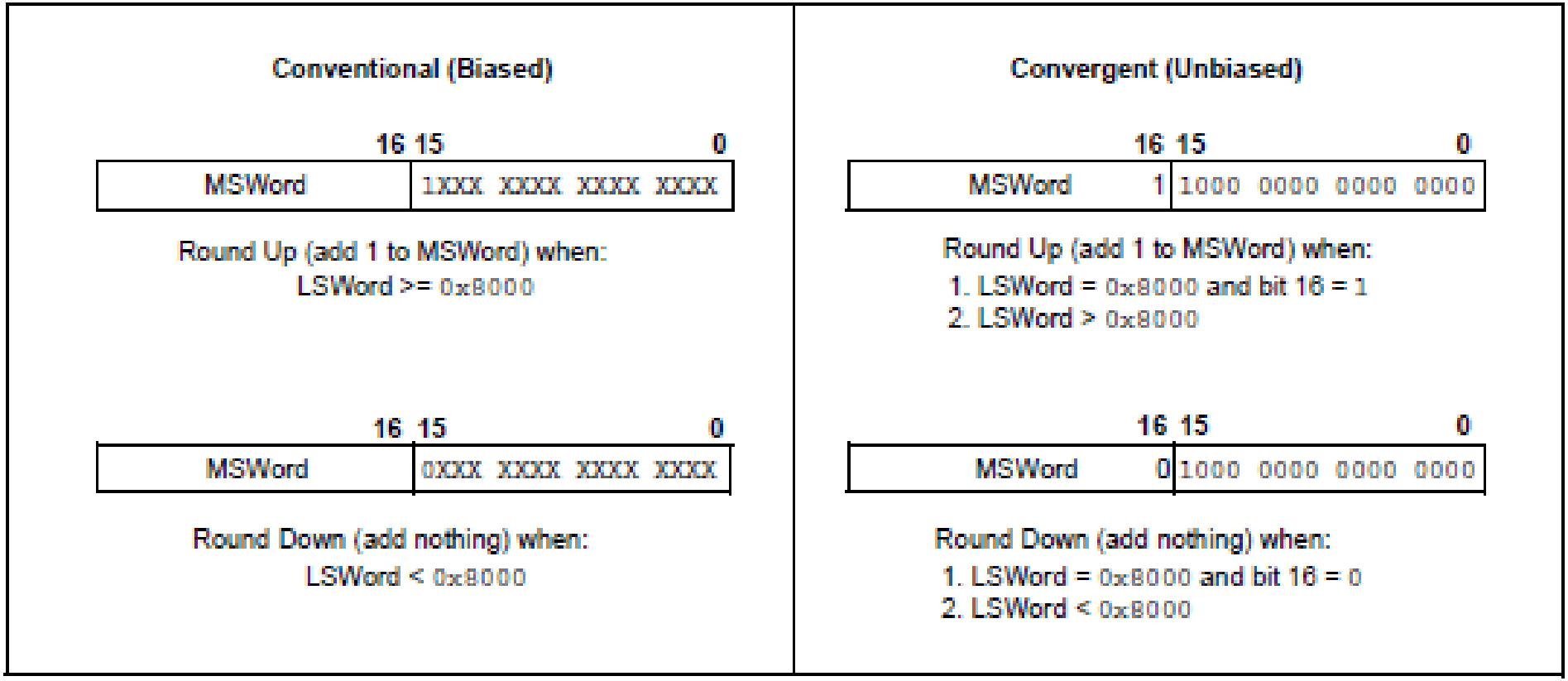
3. **Accumulator Catastrophic Overflow:** If the SATA and/or SATB bits are not set, then no saturation operation is performed on the accumulator and the accumulator is allowed to overflow all the way up to bit 39 (destroying its sign).

DSP Engine: Round Logic

- The round logic can perform a conventional (biased means result will be slightly positive) or convergent (unbiased) round function during an accumulator write (store).
- The Round mode is determined by the state of the RND (CORCON<1>) bit.
- It generates a 16-bit, 1.15 data value, which is passed to the data space write saturation logic.
- If rounding is not indicated by the instruction, a truncated 1.15 data value is stored.

DSP Engine: Round Logic

Figure 2-11: Conventional and Convergent Rounding Modes



DSP Engine: Barrel Shifter

- The barrel shifter is capable of performing up to a 16-bit arithmetic right shift, or up to a 16-bit left shift, in a single cycle.
- The barrel shifter can be used by DSP instructions or MCU instructions for multi-bit shifts.
- The shifter requires a signed binary value to determine both the magnitude (number of bits) and direction of the shift operation:
  - A positive value will shift the operand right
  - A negative value will shift the operand left
  - A value of '0' will not modify the operand
- The barrel shifter is 40-bits wide to accommodate the width of the accumulators.

DSP Engine: Barrel Shifter

- A 40-bit output result is provided for DSP shift operations, and a 16-bit result for MCU shift operations.

DSP Engine Mode Selection

➢The various operational characteristics of the DSP engine can be selected through the CPU Core Configuration register (CORCON).

- Fractional or integer multiply operation.
- Conventional or convergent rounding.
- Automatic saturation on/off for ACCA.
- Automatic saturation on/off for ACCB.
- Automatic saturation on/off for writes to data memory.
- Accumulator Saturation mode selection.

Summary

- DsPIC Architecture
- Programming Model
- DSP Engine