VERSA MIX: MIXED-SIGNAL, FULLY INTEGRATED MICROCONTROLLER WITH DSP

Datasheet Rev 1.1

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Table of Contents

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Overview

The VERSA MIX is a fully integrated mixed-signal microcontroller that provides a “one-chip solution” for a broad range of signal conditioning, data acquisition, processing, and control applications. The VERSA MIX is based on a powerful single-cycle, RISC-based, 8051 microprocessor and an enhanced MAC unit that can be used to perform complex mathematical operations.

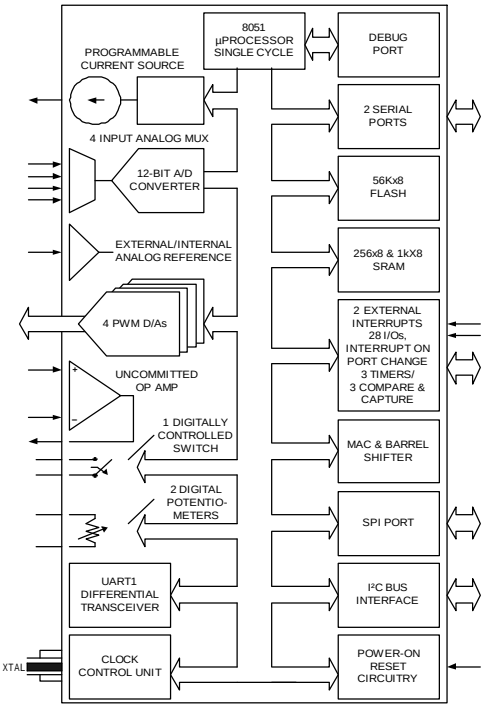
On-chip analog peripherals such as a 4-channel A/D converter, 4 PWM based D/A converters, a voltage reference, a programmable current source, an uncommitted operational amplifier, software programmable digital potentiometers and programmable switch make the VERSA MIX ideal for analog data acquisition.

The inclusion of a full set of digital interfaces such as an enhanced fully configurable SPI, an I^2C interface, UARTs and a J1708/RS-485/RS-422 compatible differential transceiver to the allow a total system integration solution on one chip.

Applications

- Automotive Applications
- Medical Devices
- Industrial Controls/ Measurement Systems
- Consumer Products
- Instrumentation
- Battery Powered SystemsIntelligent Sensors (IEEE 1451.2 Compatible)

FIGURE 1: VERSA MIX BLOCK DIAGRAM

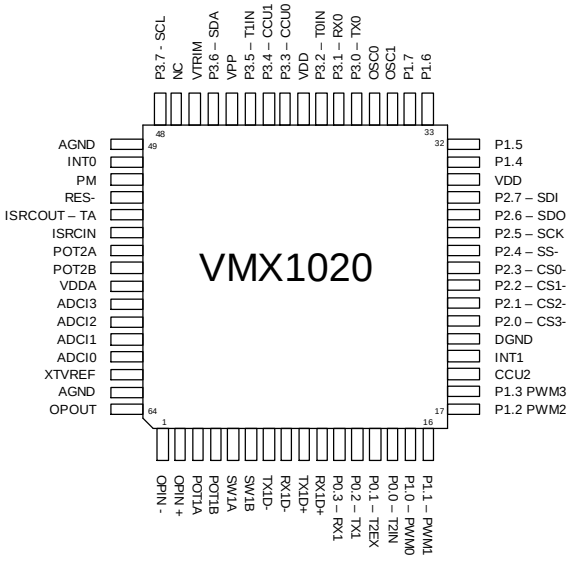


Feature Set

- 8051 Compatible RISC performances µProcessor
- Single Cycle Instructions
- Dual Data Pointers
- MAC including Barrel Shifter
- Provides DSP Capability
- 56KB Flash Program Memory
- 1280 Bytes of SRAM
- 2 Serial Interface UARTs
- Differential Transceiver connected to UART1 J1708/RS-485/RS-422 compatible.
- Enhanced SPI interface (Master/Slave)
  - Fully configurable
  - Controls up to 4 slave devices
- 28 General Purpose I/Os
- 2 General Purpose External Interrupt Inputs
- Interrupt on port 1 change
- 3 Timers / Counters
- 3 Capture and Compare Inputs
- 4-Channel 12-bit A/D Converter
  - 10kHz Conversion rate (1ch)
  - 0-2.7 Volt Input Range Continuous/One-Shot operation Single or 4-channel conversions
  - Programmable Interrupt to Processor
- On-Chip Voltage Reference
- 4 Pulse Width Modulated D/A Converters
- Programmable Current Source
- Uncommitted Operational Amplifier
- 2 Digital Potentiometers
- 1 Digitally Controlled Switch
- Power Saving Features
- Power-on Reset with Brown-Out Detect

FIGURE 2: QFP-64 PACKAGE #1 PINOUT

Pins Description for QFP-64

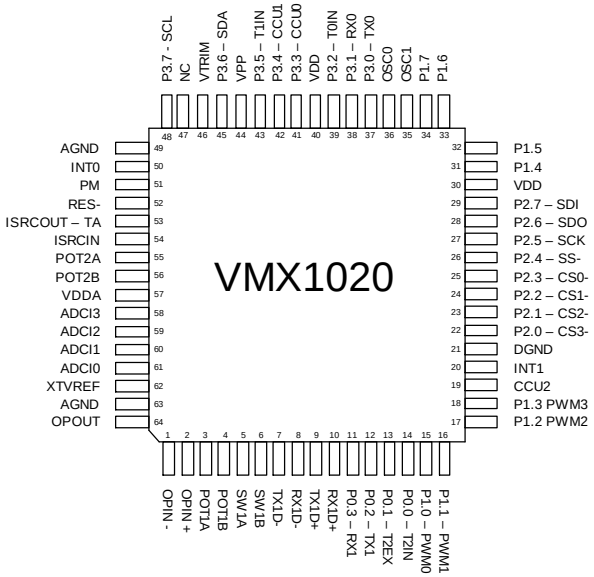


1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Table 1: Pin out description

PIN	NAME	FUNCTION
1	OPIN-	Inverting Input of the Operational Amplifier
2	OPIN+	Non-inverting Input of the Operational Amplifier
3	POT1A	Digitally Controlled Potentiometer 1A
4	POT1B	Digitally Controlled Potentiometer 1B
5	SW1A	Digitally Controlled Switch 1A
6	SW1B	Digitally Controlled Switch 1B
7	TX1D-	RS-485/RS422 compatible differential Transmitter , Negaitive side
8	RX1D-	RS-485/RS422 compatible differential Receiver Negaitive side
9	TX1D+	RS-485/RS422 compatible differential Transmitter, Positive side
10	RX1D+	RS-485/RS422 compatible differential Receiver Positive side
11	P0.3-RX1	I/0 - Asynchronous UART1 Receiver Input
12	P0.2-TX1	I/0 - Asynchronous UART1 Transmitter Output
13	P0.1-T2EX	I/0 -Timer/Counter 2 Input
14	P0.0-T2IN	I/0 -Timer/Counter 2 Input
15	P1.0-PWM0	I/0 - Pulse Width Modulated Digital to Analog Converter 0
16	P1.1-PWM1	Pulse Width Modulated Digital to Analog Converter 1
17	P1.2-PWM2	Pulse Width Modulated Digital to Analog Converter 2
18	P1.3-PWM3	Pulse Width Modulated Digital to Analog Converter 3
19	CCU2	Capture and Compare Unit 2 Input
20	INT1	Interrupt Input 1
21	DGND	Digital Ground
22	P2.0-CS3-	I/0 - SPI Chip Enable Output (Master Mode)
23	P2.1-CS2-	I/0 - SPI Chip Enable Output (Master Mode)
24	P2.2-CS1-	I/0 - SPI Chip Enable Output (Master Mode)
25	P2.3-CS0-	I/0 - SPI Chip Enable Output (Master Mode)
26	P2.4-SS-	I/0 - SPI Chip Enable Output (Slave Mode)
27	P2.5-SCK	I/0 - SPI Clock (Input in Slave Mode)
28	P2.6-SDO	I/0 - SPI Data Output Bus
29	P2.7-SDI	I/0 - SPI Data Input Bus
30	VDD	Digital Supply
31	P1.4	I/0
32	P1.5	I/0
33	P1.6	I/0
34	P1.7	I/0
35	OSC1	Oscillator Crystal Output
36	OSC0	Oscillator Crystal input/External Clock Source Input
37	P3.0-TX0	I/0 - Asynchronous UART0 Transmitter Output
38	P3.1-RX0	I/0 - Asynchronous UART0 Receiver Input
39	P3.2-T0IN	I/0 - Timer/Counter 0 Input
40	VDD	5V Digital
41	P3.3-CCU0	I/0 - Capture and Compare Unit 0 Input
42	P3.4-CCU1	I/0 - Capture and Compare Unit 1 Input
43	P3.5-T1IN	I/0 - Timer/Counter 1 Input
44	VPP	Flash Programming Voltage Input
45	P3.6-SDA	I/0 - I2C Bi-Directional Data Bus
46	VTRIM	Reserved
47	NC	Not Connected
48	P3.7-SCL	I/0 - I2C Clock
49	AGND	Analog Ground
50	INT0	External interrupt Input (Negative Level or Edge Triggered)
51	PM	Mode Control Input
52	RES-	Hardware Reset Input
53	ISRCOUT-TA	Programmable Current Source Analog Output
54	ISRCIN	Programmable Current Source Input
55	POT2A	Digitally Controlled Potentiometer 2A
56	POT2B	Digitally Controlled Potentiometer 2B
57	VDDA	Analog Supply
58	ADCI3	Analog to Digital Converter 3 Input
59	ADCI2	Analog to Digital Converter 2 Input
60	ADCI1	Analog to Digital Converter 1 Input
61	ADCI0	Analog to Digital Converter 0 Input
62	XTVREF	External Reference Voltage Input
63	AGND	Analog Ground
64	OPOUT	Output of the Operational Amplifier

FIGURE 3: VERSA MIX PINOUT OF QFP-64-1 PACKAGE



1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Absolute Maximum Ratings

V_{DD} to DGND	–0.3V, +6V	Digital Output Voltage to DGND	–0.3V, V_{DD}+0.3V
V_{DDA} to DGND	-0.3V, +6V	V_{PP} to DGND	+13V
AGND to DGND	–0.3V, +0.3V	Power Dissipation	
V_{DD} to V_{DDA}	-0.3V, +0.3V	✍ To +75°C	1000mW
ADCI(0-3) to AGND	-0.3V, V_{DDA}+0.3V	✍ Derate above +75°C	10mW/°C
XTVREF to AGND	-0.3V, V_{DDA}+0.3V	Operating Temperature range	-40° to +85°C
Digital Input Voltage to DGND	-0.3V, V_{DD}+0.3V	Storage Temperature Range	–65°C to +150°C
RS422/485 Minimum and maximum Voltages	-2V, +7V	Lead Temperature (soldering, 10sec)	+300°C

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Electrical Characteristics

TABLE 2: ELECTRICAL CHARACTERISTICS

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
GENERAL CHARACTERISTICS (V_{DD} = +5V, V_{DDA} = +5V, T_A = +25°C, 16MHz input clock, unless otherwise noted.)						
Power Supply Voltage	V_{DD}		4.5	5.0	5.5	V
	V_{DDA}		4.5	5.0	5.5	V
Power Supply Current	I_{DD}		5		50*	mA
	I_{DDA}		0.1		5*	* Depends on clock speed and peripheral use
Flash Programming Voltage	V_{PP}		11		13	V
DIGITAL INPUTS						
Minimum High-Level Input Voltage	V_{IH}	V_{DD} = +5V		2.0		V
Maximum Low-Level Input Voltage	V_{IL}	V_{DD} = +5V		0.8		V
Input Current	I_{IN}			±0.05		µA
Input Capacitance	C_{IN}			5	10	pF
DIGITAL OUTPUTS						
Minimum High-Level Output Voltage	V_{OH}	I_{SOURCE} = 4mA		4.2		V
Maximum Low-Level Output Voltage	V_{OL}	I_{SINK} = 4mA		0.2		V
Output Capacitance	C_{OUT}			10	15	pF
Tri-state Output Leakage Current	I_{OZ}				0.25	µA

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
ANALOG INPUTS						
ADCI(0-3) Input Voltage Range	V_{ADCI}		0		2.7	V
ADCI(0-3) Input Resistance	R_{ADCI}			100		Mohms (design)
ADCI(0-3) Input Capacitance	C_{ADCI}			7		pF
ADCI(0-3) Input Leakage Current	I_{ADCI}			TBD		nA
Channel-to-Channel Crosstalk					-72 (12 bit)	dB (design)
ANALOG OUTPUT						
TA Output Drive Capabilities (Maximum Load Resistance)	$V_{TA=}V_{ADCI(0-3)}$		10			kOhms
	Others	Requires buffering		25M		
CURRENT SOURCE						
ISRC Current Drive	I_{ISRC}		0		530	µA
ISRC Feedback voltage 200mV	$REF_{ISRC200}$		195		205	mV
ISRC Feedback voltage 800mV	$REF_{ISRC800}$		799		803	mV
ISRC Output Resistance	R_{ISRC}			50		MOhms
ISRC Output Capacitance	C_{ISRC}			25		pF
ISRCIN Input Reference Resistance	R_{RESIN}			100		M?
ISRCIN Input Reference Capacitance	C_{RESIN}			7		pF
ISRC stability	Drift				2.5	%
INTERNAL REFERENCE						
Bandgap Reference Voltage			1.18	1.23V	1.28	V
Bandgap Reference Tempco				100		ppm/°C
EXTERNAL REFERENCE						
Input Impedance	R_{XTVREF}			150		kOhms
PGA						
PGA Gain adjustment			2.11		2.29	V

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
ANALOG TO DIGITAL CONVERTER **External Reference, TA=25C, Fosc = 16MHz**						
ADC Resolution				12		Bits
Differential Non linearity	DNL				±1.5	LSB
Integral Non linearity	INL		-1		+4	LSB
Full-Scale Error (Gain Error)		All channels, ADCI(0-3)		±4		LSB
Offset Error		All channels, ADCI(0-3)		±1		LSB
Channel-to-Channel Mismatch		All channels, ADCI(0-3)		±1		LSB
Sampling Rate		Single Channel	1		10	kHz
		4 Channels	1		2.5	
UART1 DIFFERENTIAL TRANSCEIVER COMPATIBLE TO J1708/ RS-485/RS-422						
Common mode Input Voltage	Vc_I		-2		+7	V
Input Impedance	Z_{IN}			1		MOhms
Output Drive Current				30		mA
Differential Input				100mV		mV

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

USER OP-AMP					
Output Impedance	Zout		20		mOhms
Input Resistance	Zin		36		GOhms
Voltage Gain	G_v		105		dB
Unit Gain Bandwidth	UGBW		5		MHz
Load Resistance to Ground			10		KOhms
Load Capacitance			18		pF
Slew rate	SR		5		V/µs (Design)
Power Dissipation	P_D	TA= 25°C	1150		mW (Design)
Input Offset Voltage	V_{IO}		+/- 2		mV
Input Voltage Range	$V_{I®}$		0	4	V (Design)
Common Mode Rejection Ratio	CMRRdc	DC	83	99	dB
	CMRR1kHz	Taken at 1kHz	75		dB Design)
Power Supply Rejection Ratio	PSRR	Taken at 1kHz (20dB/decade)	-75 (Vdd)	-94 (Vss)	dB (Design)
Output Voltage Swing (RL=10k)	$V_{O(P-P)}$		25mV	4.975	V (Design)
Short Circuit Current to ground	I_{IC}		86		MA (Design)
Gain Bandwidth Product	GBW		525		MdB (Design)
DIGITAL POTENTIOMETERS					
Number of Steps (8 bit binary weighted)			256		steps
Resistance			125	28	Ohms (+/- 15%)
Absolute Linearity			TBD		%
Interchannel Matching			1		%
Temperature Coefficient			0.16		%/°C
Inherent Capacitance			3		pF
DIGITAL SWITCH					
Switch on Resistance			50	100	Ohms (+/-10%)
Input capacitance			4		pF
Voltage range on Pin			0	5	V

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Detailed Description

The following sections will describe the VERSA MIX architecture and peripherals.

FIGURE 4: INTERFACE DIAGRAM FOR THE VERSA MIX

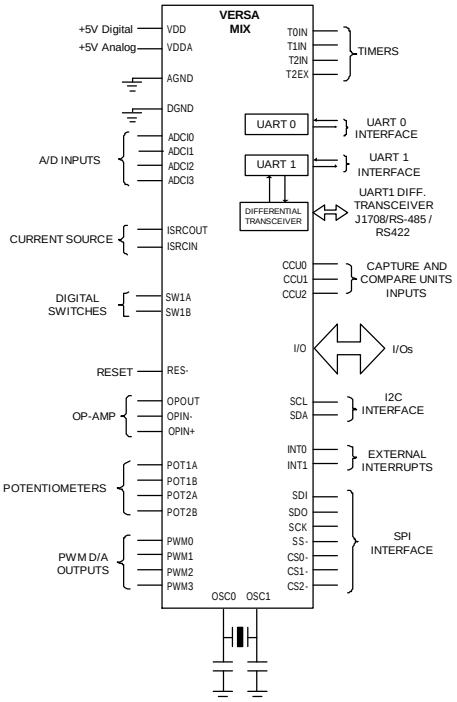
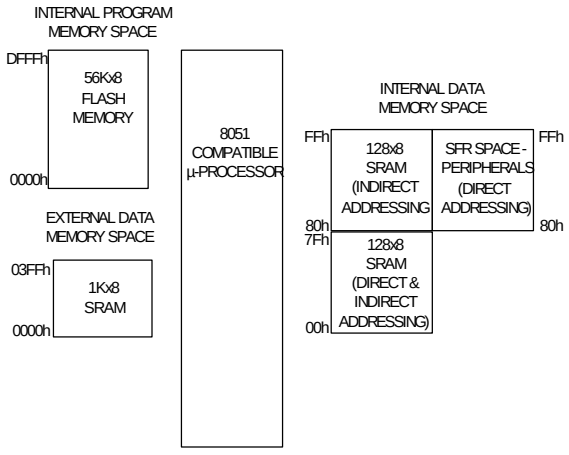


FIGURE 5: MEMORY ORGANIZATION OF THE VERSA MIX



Memory Organization

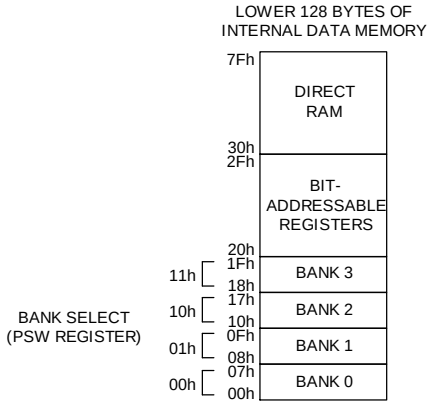
Figure 5 shows the memory organization of the VERSA MIX.

At power-up/reset, the code is executed from the 56Kx8 Flash memory mapped into the processor's internal ROM space.

A 1Kx8 block of SRAM is also mapped into the external data memory of the VERSA MIX. This block can be used as a general-purpose scratch pad or storage memory. A 256x8 block of SRAM is mapped to the internal data memory space. This block of RAM is broken into 2 sub-blocks, with the upper block accessible via indirect addressing only, and the lower block accessible via both direct and indirect addressing.

The following figure describes the access to the lower block of 128 bytes.

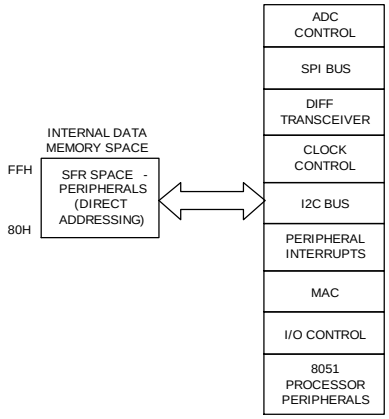
FIGURE 6: LOWER 128 BYTES BLOCK INTERNAL MEMORY MAP



The SFR (Special Function Register) space is also mapped into the upper 128 bytes of internal data memory space. This SFR space is only accessible using direct-access. The SFR space provides the interface to all the on-chip peripherals. This interfacing is illustrated in figure 7.

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

FIGURE 7: SFR ORGANIZATION



Dual Data Pointers

The VERSA MIX employs dual data pointers to accelerate data memory block moves. The standard 8051 data pointer (DPTR) is a 16-bit value used to address data RAM or program memory. The VERSA MIX maintains the standard data pointer as DPTR0 at SFR locations 82h and 83h.

The VERSA MIX adds a second data pointer (DPTR1) at SFR Locations 84h and 85h. The SEL bit in the data pointer select register, DPS (SFR 86h), selects which data pointer is active. When SEL = 0, instructions that use the data pointer will use DPL0 and DPH0. When SEL = 1, instructions that use the DPTR will use DPL1 and DPH1. SEL is the bit 0 of SFR location 86h. No other bits of SFR location 86h are used.

All DPTR-related instructions use the currently selected data pointer. In order to switch the active pointer, toggle the SEL bit. The fastest way to do so is to use the increment instruction (INC DPS). This requires only one instruction to switch from a source address and destination address. This will prevent application code from having to save source and destination addresses when doing a block move.

The use of the two data pointers can significantly increases the speed of moving large blocks of data. The SFR locations and register representations related to the dual data pointers are:

TABLE 3: (DPH0) DATA POINTER HIGH 0 - SFR 83H

15	14	13	12	11	10	9	8
DPH0 [7:0]							

TABLE 4: (DPL0) DATA POINTER LOW 0 - SFR 82H

7	6	5	4	3	2	1	0
DPL0 [7:0]							

Bit	Mnemonic	Function
15-8	DPH0	Data pointer high 0. Used to access external code or data space.
7-0	DPL0	Data pointer low 0. Used to access external code or data space.

TABLE 5: (DPH1) DATA POINTER HIGH 1 - SFR 85H

15	14	13	12	11	10	9	8
DPH1 [7:0]							

TABLE 6: (DPL1) DATA POINTER LOW 1 - SFR 84H

7	6	5	4	3	2	1	0
DPL1 [7:0]							

Bit	Mnemonic	Function
15-8	DPH1	Data pointer high 1. Used to access external code or data space.
7-0	DPL1	Data pointer low 1. Used to access external code or data space.

TABLE 7: (DPS) DATA POINTER SELECT REGISTER - SFR 86H

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	SEL

Bit	Mnemonic	Function
7-1	0	Always zero
0	SEL	Used to toggle between both data pointers

MPAGE Register

The MPAGE register controls the upper 8 bits when we perform the MOVX instruction. It is used for external RAM jump control.

TABLE 8: (MPAGE) MEMORY PAGE - SFR CFH

7	6	5	4	3	2	1	0
MPAGE [7:0]							

User Flags

The VERSA MIX provides an SFR register that gives the user the ability to define software flags. It is for this purpose that each bit in this register is individually addressable. This register may also be used as a general-purpose storage location. Thus, the user flag feature allows the VERSA MIX to better adapt to each specific application. This register is located at SFR address F8h

TABLE 9: (USERFLAGS) USER FLAG - SFR F8H

7	6	5	4	3	2	1	0
TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Instruction Set

All VERSA MIX instructions are binary code compatible and perform the same functions as the industry standard 8051. However, the timing of the instructions is different. The following two tables describe the instruction set of the VERSA MIX.

TABLE 10: LEGEND FOR INSTRUCTION SET TABLE

Symbol	Function
A	Accumulator
Rn	Register R0-R7
Direct	Internal register address
@Ri	Internal register pointed to by R0 or R1 (except MOVX)
rel	Two's complement offset byte
bit	Direct bit address
#data	8-bit constant
#data 16	16-bit constant
addr 16	16-bit destination address
addr 11	11-bit destination address

TABLE 11: VERSA MIX INSTRUCTION SET

Mnemonic	Description	Size (bytes)	Instr. Cycles
Arithmetic instructions			
ADD A, Rn	Add register to A	1	1
ADD A, direct	Add direct byte to A	2	2
ADD A, @Ri	Add data memory to A	1	2
ADD A, #data	Add immediate to A	2	2
ADDC A, Rn	Add register to A with carry	1	1
ADDC A, direct	Add direct byte to A with carry	2	2
ADDC A, @Ri	Add data memory to A with carry	1	2
ADDC A, #data	Add immediate to A with carry	2	2
SUBB A, Rn	Subtract register from A with borrow	1	1
SUBB A, direct	Subtract direct byte from A with borrow	2	2
SUBB A, @Ri	Subtract data mem from A with borrow	1	2
SUBB A, #data	Subtract immediate from A with borrow	2	2
INC A	Increment A	1	1
INC Rn	Increment register	1	2
INC direct	Increment direct byte	2	3
INC @Ri	Increment data memory	1	3
DEC A	Decrement A	1	1
DEC Rn	Decrement register	1	2
DEC direct	Decrement direct byte	2	3
DEC @Ri	Decrement data memory	1	3
INC DPTR	Increment data pointer	1	1
MUL AB	Multiply A by B	1	5
DIV AB	Divide A by B	1	5
DA A	Decimal adjust A	1	1
Logical Instructions			
ANL A, Rn	AND register to A	1	1
ANL A, direct	AND direct byte to A	2	2
ANL A, @Ri	AND data memory to A	1	2
ANL A, #data	AND immediate to A	2	2
ANL direct, A	AND A to direct byte	2	3
ANL direct, #data	AND immediate data to direct byte	3	4
ORL A, Rn	OR register to A	1	1
ORL A, direct	OR direct byte to A	2	2
ORL A, @Ri	OR data memory to A	1	2
ORL A, #data	OR immediate to A	2	2
ORL direct, A	OR A to direct byte	2	3
ORL direct, #data	OR immediate data to direct byte	3	4
XRL A, Rn	Exclusive-OR register to A	1	1
XRL A, direct	Exclusive-OR direct byte to A	2	2
XRL A, @Ri	Exclusive-OR data memory to A	1	2
XRL A, #data	Exclusive-OR immediate to A	2	2
XRL direct, A	Exclusive-OR A to direct byte	2	3
XRL direct, #data	Exclusive-OR immediate to direct byte	3	4
CLR A	Clear A	1	1
CPL A	Compliment A	1	1
SWAP A	Swap nibbles of A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through carry	1	1
Data Transfer Instructions			
MOV A, Rn	Move register to A	1	1
MOV A, direct	Move direct byte to A	2	2
MOV A, @Ri	Move data memory to A	1	2
MOV A, #data	Move immediate to A	2	2
MOV Rn, A	Move A to register	1	2
MOV Rn, direct	Move direct byte to register	2	4
MOV Rn, #data	Move immediate to register	2	2
MOV direct, A	Move A to direct byte	2	3
MOV direct, Rn	Move register to direct byte	2	3
MOV directa, directb	Move directb byte to directa byte	3	4
MOV direct, @Ri	Move data memory to direct byte	2	4
MOV direct, #data	Move immediate to direct byte	3	3
MOV @Ri, A	Move A to data memory	1	3
MOV @Ri, direct	Move direct byte to data memory	2	5
MOV @Ri, #data	Move immediate to data memory	2	3
MOV DPTR, #data	Move immediate to data pointer	3	3
MOVC A, @A+DPTR	Move code byte relative DPTR to A	1	3
MOVC A, @A+PC	Move code byte relative PC to A	1	3
MOVX A, @Ri	Move external data (A8) to A	1	3-10
MOVX A, @DPTR	Move external data (A16) to A	1	3-10
MOVX @Ri, A	Move A to external data (A8)	1	4-11
MOVX @DPTR, A	Move A to external data (A16)	1	4-11
PUSH direct	Push direct byte onto stack	2	4
POP direct	Pop direct byte from stack	2	3
XCH A, Rn	Exchange A and register	1	2
XCH A, direct	Exchange A and direct byte	2	3
XCH A, @Ri	Exchange A and data memory	1	3
XCHD A, @Ri	Exchange A and data memory nibble	1	3
Branching Instructions			
ACALL addr 11	Absolute call to subroutine	2	6
LCALL addr 16	Long call to subroutine	3	6
RET	Return from subroutine	1	4
RETI	Return from interrupt	1	4
AJMP addr 11	Absolute jump unconditional	2	3
LJMP addr 16	Long jump unconditional	3	4
SJMP rel	Short jump (relative address)	2	3
JC rel	Jump on carry = 1	2	3
JNC rel	Jump on carry = 0	2	3
JB bit, rel	Jump on direct bit = 1	3	4
JNB bit, rel	Jump on direct bit = 0	3	4
JBC bit, rel	Jump on direct bit = 1 and clear	3	4
JMP @A+DPTR	Jump indirect relative DPTR	1	2
JZ rel	Jump on accumulator = 0	2	3
JNZ rel	Jump on accumulator 1= 0	2	3
CJNE A, direct, rel	Compare A, direct JNE relative	3	4
CJNE A, #data, rel	Compare A, immediate JNE relative	3	4
CJNE Rn, #data, rel	Compare reg, immediate JNE relative	3	4
CJNE @Ri, #data, rel	Compare ind, immediate JNE relative	3	4
DJNZ Rn, rel	Decrement register, JNZ relative	2	3
DJNZ direct, rel	Decrement direct byte, JNZ relative	3	4
Bit Operations			
CLR C	Clear carry flag	1	1
CLR bit	Clear direct bit	2	3
SET C	Set carry flag	1	1
SET bit	Set direct bit	2	3
CPL C	Complement carry Flag	1	1
CPL bit	Complement direct bit	2	3
ANL C,bit	Logical AND direct bit to carry flag	2	2
ANL C, /bit	Logical AND between /bit and carry flag	2	2
ORL C,bit	Logical OR bit to carry flag	2	2
ORL C, /bit	Logical OR /bit to carry flag	2	2
MOC c,bit	Copy direct bit location to carry flag	2	2
MOV bit,C	Copy carry flag to direct bit location	2	3
Miscellaneous Instruction			
NOP	No operation	1	1

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Special Function Registers

The Special Function Registers (SFRs) control several of the features of the VERSA MIX. Many of the VERSA MIX SFRs are identical to the standard 8051 SFRs. However, there are additional SFRs that control the VERSA MIX's specific peripheral features that are not available in the standard 8051.

TABLE 12: SPECIAL FUNCTION REGISTERS

SFR Register	SFR Adrs	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Value
P0	80h	-	-	-	-	-	-	-	-	1111 1111b
SP	81h	-	-	-	-	-	-	-	-	0000 0111b
DPL0	82h	-	-	-	-	-	-	-	-	0000 0000b
DPH0	83h	-	-	-	-	-	-	-	-	0000 0000b
DPL1	84h	-	-	-	-	-	-	-	-	0000 0000b
DPH1	85h	-	-	-	-	-	-	-	-	0000 0000b
DPS	86h	0	0	0	0	0	0	0	SEL	0000 0000b
PCON	87h	SMOD_0	-	-	-	GF1	GF0	STOP	IDLE	0000 0000b
TCON*	88h	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	0000 0000b
TMOD	89h	GATE1	CT1	M11	M01	GATE0	CT_0	M10	M00	0000 0000b
TL0	8Ah	-	-	-	-	-	-	-	-	0000 0000b
TL1	8Bh	-	-	-	-	-	-	-	-	0000 0000b
TH0	8Ch	-	-	-	-	-	-	-	-	0000 0000b
TH1	8Dh	-	-	-	-	-	-	-	-	0000 0000b
Reserved	8Eh									
Reserved	8Fh									
P1*	90h	-	-	-	-	-	-	-	-	1111 1111b
IRCON	91h	EXF2IF	TF2IF	ADCIF	MACIF	I2CIF	SPIRXIF	SPITXIF	Reserved	0000 0000b
ANALOGPWREN	92h	OPAMPEN	DIGPOTEN	ISRCSEL	ISRCEN	TAEN	ADCEN	PGAEN	BGAPEN	0000 0000b
DIGPWREN	93h	T2CLKEN	WDOGEN	MACEN	I2CEN	SPIEN	UART1DIFFEN	UART1EN	UART0EN	0000 0000b
CLKDIVCTRL	94h	SOFTRST	-	-	IRQNORMSPD	MCKDIV_3	MCKDIV_2	MCKDIV_1	MCKDIV_0	0000 0000b
ADCCLKDIV	95h	-	-	-	-	-	-	-	-	0000 0000b
S0RELL	96h	-	-	-	-	-	-	-	-	11011001b
S0RELH	97h	0	0	0	0	0	0	-	-	0000 0011b
S0CON*	98h	S0M0	S0M1	MCPE0	R0EN	T0B8	R0B8	T0I	R0I	0000 0000b
S0BUF	99h	-	-	-	-	-	-	-	-	0000 0000b
IEN2	9Ah	-	-	-	-	-	-	-	S1IE	0000 0000b
P0PINCFG	9Bh	P07IO	P06IO	P05IO	P04IO	P0.3/RX1IE	P0.2/TX1IE	P0.1/T2EXIE	P0.0/T2INE	0000 0000b
P1PINCFG	9Ch	P1.7EN	P1.6EN	P1.5EN	P1.4EN	PWM3EN	PWM2EN	PWM1EN	PWM0EN	
P2PINCFG	9Dh	SDIEN	SDOEN	SCKEN	SSEN	CS0EN	CS1EN	CS2EN	CS3EN	0000 0000b
P3PINCFG	9Eh	MSCLEN	MSDAEN	T1INEN	CCU1EN	CCU0EN	T0INEN	RX0EN	TX0EN	0000 0000b
PORTIRQEN	9Fh	P07IEN	P06IEN	P05IEN	P04IEN	P03IEN	P02IEN	P01IEN	P00IEN	0000 0000b
P2*	A0h	-	-	-	-	-	-	-	-	1111 1111b
PORTIRQSTAT	A1h	P07ISTAT	P06ISTAT	P05ISTAT	P04ISTAT	P03ISTAT	P02ISTAT	P01ISTAT	P00ISTAT	0000 0000b
ADCCTRL	A2h	ADCIRQCLR	XVREFCAP	1	ADCIRQ	ADCIE	ONECHAN	CONT	ONESHOT	0000 0000b
ADCCONVRLOW	A3h	-	-	-	-	-	-	-	-	0000 0000b
ADCCONVRMED	A4h	-	-	-	-	-	-	-	-	0000 0000b
ADCCONVRHIGH	A5h	-	-	-	-	-	-	-	-	0000 0000b
ADCD0LO	A6h	-	-	-	-	-	-	-	-	0000 0000b
ADCD0HI	A7h					ADCD0HI_3	ADCD0HI_2	ADCD0HI_1	ADCD0HI_0	0000 0000b
IEN0*	A8h	EA	WDT	T2IE	S0IE	T1IE	INT1IE	T0IE	INT0IE	0000 0000b
ADCD1LO	A9h	-	-	-	-	-	-	-	-	0000 0000b
ADCD1HI	AAh					ADCD1HI_3	ADCD1HI_2	ADCD1HI_1	ADCD1HI_0	0000 0000b
ADCD2LO	ABh	-	-	-	-	-	-	-	-	0000 0000b
ADCD2HI	ACh					ADCD2HI_3	ADCD2HI_2	ADCD2HI_1	ADCD2HI_0	0000 0000b
ADCD3LO	ADh	-	-	-	-	-	-	-	-	0000 0000b
ADCD3HI	AEh					ADCD3HI_3	ADCD3HI_2	ADCD3HI_1	ADCD3HI_0	0000 0000b
Reserved	AFh									
P3*	B0h	-	-	-	-	-	-	-	-	1111 1111b
Reserved	B1h									
Reserved	B2h									
BGAPCAL	B3h	-	-	-	-	-	-	-	-	0000 0000b
PGACAL	B4h	-	-	-	-	-	-	-	-	0000 0000b
INMUXCTRL	B5h	NOT USED	ADCINSEL_2	ADCINSEL_1	ADCINSEL_0	AINEN_3	AINEN_2	AINEN_1	AINEN_0	0000 0000b
OUTMUXCTRL	B6h	-	-	-	-	-	TAOUTSEL_2	TAOUTSEL_1	TAOUTSEL_0	0000 0000b
SWITCHCTRL	B7h	-	-	-	-	SWITCH1_3	SWITCH1_2	SWITCH1_1	SWITCH1_0	0000 0000b
IP0*	B8h	UF8	WDTS	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0	0000 0000b
IP1	B9h	-	-	IP1.5	IP1.4	IP1.3	IP1.2	IP1.1	IP1.0	0000 0000b
DIGPOT1	BAh	-	-	-	-	-	-	-	-	0000 0000b
DIGPOT2	BBh	-	-	-	-	-	-	-	-	0000 0000b

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

SFR Register	SFR Adrs	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Value
ISRCCAL1	BCh	PGACAL0	ISRCCAL1_6	ISRCCAL1_5	ISRCCAL1_4	ISRCCAL1_3	ISRCCAL1_2	ISRCCAL1_1	ISRCCAL1_0	0000 0000b
ISRCCAL2	BDh	-	ISRCCAL2_6	ISRCCAL2_5	ISRCCAL2_4	ISRCCAL2_3	ISRCCAL2_2	ISRCCAL2_1	ISRCCAL2_0	0000 0000b
S1RELL	BEh	-	-	-	-	-	-	-	-	0000 0000b
S1RELH	BFh	-	-	-	-	-	-	-	-	0000 0000b
S1CON*	C0h	S1M	reserved	MCPE1	R1EN	T1B8	R1B8	T1I	R1I	0000 0000b
S1BUF	C1h	-	-	-	-	-	-	-	-	0000 0000b
CCL1	C2h	-	-	-	-	-	-	-	-	0000 0000b
CCH1	C3h	-	-	-	-	-	-	-	-	0000 0000b
CCL2	C4h	-	-	-	-	-	-	-	-	0000 0000b
CCH2	C5h	-	-	-	-	-	-	-	-	0000 0000b
CCL3	C6h	-	-	-	-	-	-	-	-	0000 0000b
CCH3	C7h	-	-	-	-	-	-	-	-	0000 0000b
T2CON*	C8h	T2PS	T2PSM	T2S	T2R1	T2R0	T2CM	T2I1	T2I0	0000 0000b
CCEN	C9h	COCAH3	COCAL3	COCAH2	COCAL2	COCAH1	COCAL1	COCAH0	COCAL0	0000 0000b
CRCL	CAh	-	-	-	-	-	-	-	-	0000 0000b
CRCH	CBh	-	-	-	-	-	-	-	-	0000 0000b
TL2	CCh	-	-	-	-	-	-	-	-	0000 0000b
TH2	CDh	-	-	-	-	-	-	-	-	0000 0000b
Reserved	CEh									
MPAGE	CFh	-	-	-	-	-	-	-	-	0000 0000b
PSW*	D0h	CY	AC	F0	RS1	RS0	OV	reserved	P	0000 0000b
Reserved	D1h									
Reserved	D2h									
Reserved	D3h									
Reserved	D4h									
Reserved	D5h									
Reserved	D6h									
Reserved	D7h									
U0BAUD	D8h	BAUDSRC	-	-	-	-	-	-	-	0000 0000b
WDTREL	D9h	PRES	WDTREL_6	WDTREL_5	WDTREL_4	WDTREL_3	WDTREL_2	WDTREL_1	WDTREL_0	0000 0000b
I2CCONFIG	DAh	I2CMASKID	I2CRXOVIE	I2CRXDAVIE	I2CTXEMPIE	I2CMANACK	I2CACKMODE	I2CMSTOP	I2CMASTER	0000 0010b
I2CCLKCTRL	DBh	-	-	-	-	-	-	-	-	0000 0000b
I2CCHIPID	DCh	I2CID_6	I2CID_5	I2CID_4	I2CID_3	I2CID_2	I2CID_1	I2CID_0	I2CWID	0100 0010b
I2CIRQSTAT	DDh	I2CGOTSTOP	I2CNOACK	I2CSDASYNC	I2CDATACK	I2CIDLE	I2CRXOV	I2CRXAV	I2CTXEMP	0010 1001b
I2CRXTX	DEh	-	-	-	-	-	-	-	-	0000 0000b
Reserved	DFh									
ACC*	E0h	-	-	-	-	-	-	-	-	0000 0000b
SPIRX3TX0	E1h	-	-	-	-	-	-	-	-	0000 0000b
SPIRX2TX1	E2h	-	-	-	-	-	-	-	-	0000 0000b
SPIRX1TX2	E3h	-	-	-	-	-	-	-	-	0000 0000b
SPIRX0TX3	E4h	-	-	-	-	-	-	-	-	0000 0000b
SPICTRL	E5h	SPICK_2	SPICK_1	SPICK_0	SPICS_1	SPICS_0	SPICKPH	SPICKPOL	SPIMA_SL	0000 0001b
SPICONFIG	E6h	SPICSLO	-	FSONCS3	SPI LOAD	NOT USED	SPIRXOVIE	SPIRXAVIE	SPITXEMPIE	0000 0000b
SPISIZE	E7h	-								0000 0111b
IEN1*	E8h	T2EXIE	SWDT	ADCPCIE	MACOVIE	I2CIE	SPIOVIE	SPITEIE	reserved	0000 0000b
SPIIRQSTAT	E9h	-	-	SPITXEMPTO	SPISLAVESEL	SPISEL	SPIOV	SPIRAXV	SPITXEMP	00011001b
Reserved	EAh									
MACCTRL1	EBh	LOADPREV	PREVMODE	OVMODE	OVRDVAL	ADDSRC_1	ADDSRC_0	MULCMD_1	MULCMD_0	0000 0000b
MACC0	ECh	-	-	-	-	-	-	-	-	0000 0000b
MACC1	EDh	-	-	-	-	-	-	-	-	0000 0000b
MACC2	EEh	-	-	-	-	-	-	-	-	0000 0000b
MACC3	EFh	-	-	-	-	-	-	-	-	0000 0000b
B*	F0h	-	-	-	-	-	-	-	-	0000 0000b
MACCTRL2	F1h	MACCLR2_2	MACCLR2_1	MACCLR2_0	MACOV32IE	-	-	MACOV16	MACOV32	0000 0000b
MACA0	F2h	-	-	-	-	-	-	-	-	0000 0000b
MACA1	F3h	-	-	-	-	-	-	-	-	0000 0000b
MACRES0	F4h	-	-	-	-	-	-	-	-	0000 0000b
MACRES1	F5h	-	-	-	-	-	-	-	-	0000 0000b
MACRES2	F6h	-	-	-	-	-	-	-	-	0000 0000b
MACRES3	F7h	-	-	-	-	-	-	-	-	0000 0000b
USERFLAGS*	F8h	UF7	UF6	UF5	UF4	UF3	UF2	UF1	UF0	0000 0000b
MACB0	F9h	-	-	-	-	-	-	-	-	0000 0000b
MACB1	FAh	-	-	-	-	-	-	-	-	0000 0000b
MACSHIFTCTRL	FBh	SHIFTMODE	ALSHSTYLE	SHIFTAMPL_5	SHIFTAMPL_4	SHIFTAMPL_3	SHIFTAMPL_2	SHIFTAMPL_1	SHIFTAMPL_0	0000 0000b
MACPREV0	FCh	-	-	-	-	-	-	-	-	0000 0000b
MACPREV1	FDh	-	-	-	-	-	-	-	-	0000 0000b
MACPREV2	FEh	-	-	-	-	-	-	-	-	0000 0000b
MACPREV3	FFh	-	-	-	-	-	-	-	-	0000 0000b

* Bit addressable

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Peripheral Interfaces

Digital peripherals Power Enable

By default, upon reset, most of the peripherals of the VERSA MIX are not activated.

In order to use a given digital peripherals, it must first be enabled by setting to 1 the corresponding bit of the DIGPWREN SFR register.

The following table shows the structure of the DIGPWREN register.

TABLE 13: (DIGPWREN) DIGITAL PERIPHERALS POWER ENABLE REGISTER - SFR 93H

7	6	5	4
T2CLKEN	WDOGEN	MACEN	I2CEN

3	2	1	0
SPIEN	UART1DIFFEN	UART1EN	UART0EN

Bit	Mnemonic	Function
7	T2CLKEN	Timer2 / PWM Enable 0 = Timer 2 CLK stopped 1 = Timer 2 CLK Running
6	WDOGEN	Watch dog Enable 0 = Watchdog Disable 1 = Watch Dog Enable
5	MACEN	1 = MAC Unit Enable 0 = MAC Unit Disable
4	I2CEN	1= I2C Interface Enable 0 = I2C Interface Disable This bit is merged with CLK STOP bit
3	SPIEN	1 = SPI interface is Enable 0 = SPI interface is Disable
2	UART1DIFFEN	UART1 Differential mode 0 = Disable 1 = Enable
1	UART1EN	0 = UART1 Disable 1 = UART1 Enable
0	UART0EN	0 = UART0 Disable 1 = UART0 Enable

Analog peripherals Power Enable

The analog peripherals such as the op-amp, digital potentiometer, current source and analog to digital converter have one register dedicated to enabling and disabling them. This register and its bit functions are found in the table below

TABLE 14: (ANALOGPWREN) ANALOG PERIPHERALS POWER ENABLE REGISTER - SFR 92H

7	6	5	4
OPAMPEN	DIGPOTEN	ISRCSEL	ISRCEN

3	2	1	0
TAEN	ADCEN	PGAEN	BGAPEN

Bit	Mnemonic	Function
7	OPAMPEN	1 = User Op-Amp Enable 0 = User Op-Amp Disable
6	DIGPOTEN	1 = Digital Potentiometer and Switch Enable 0 = Digital Potentiometer and Switch Disable
5	ISRCSEL	0 = ISRC with 200mV feedback 1 = ISRC with 200mV feedback
4	ISRCEN	1 = ISRC Output Enable 0 = ISRC Output Disable
3	TAEN	1 = TA Output Enable 0 = TA Output Disable
2	ADCEN	1 = ADC Enable 0 = ADC Disable
1	PGAEN	1 = PGA Enable 0 = PGA Disable
0	BGAPEN	1 = Bandgap Enable 0 = Bandgap Disable

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

General Purpose I/O

The VRS1020 devices can be configured to provide up to 28 I/O pins. The I/Os are shared with the digital peripherals and can be configured individually.

Ports are bi-directional. This means, the CPU can output or read data through any of these ports.

TABLE 15:
PORT 0 - SFR 80H

7	6	5	4	3	2	1	0
P0 [7:0]							

PORT 1 - SFR 90H

7	6	5	4	3	2	1	0
P1 [7:0]							

PORT 2 - SFR A0H

7	6	5	4	3	2	1	0
P2 [7:0]							

PORT 3 - SFR B0H

7	6	5	4	3	2	1	0
P3 [7:0]							

Bit	Mnemonic	Function
7-0	P0,1,2,3	When the Port is configured as an output, Setting a port pin to 1 will make the corresponding pin to output logic high. When set to 0, the corresponding pin will set a logic low.

I/O Ports configuration registers

The following registers are used to configure each of the ports as either a general purpose inputs, outputs or alternate peripheral function.

The port pin configuration combined with specific peripheral configuration will define if a given pin act as a general purpose I/O or if provide the alternate peripheral functionality.

Before using a peripheral shared with I/Os you must make sure that the pin corresponding to the peripherals output are configured as output and that pins that are shared with the peripheral inputs are configured as input.

For example, when bit 5 of Port 2 is configured as an output, it will output the SCK signal if the SPI interface is enabled and working.

The only exception to this rule is the I^2C Clock and data bus signals. In these two cases, the VERSA MIX configures the pins automatically as inputs or outputs.

TABLE 16: (P0PINCFG) PORT 0 PORT CONFIGURATION REGISTER - SFR 9BH

7	6	5	4
P07IO	P06IO	P05IO	P04IO

3	2	1	0
P0.3/RX1IE	P0.2/TX1OE	P0.1/T2EXOE	P0.0/T2INOE

Bit	Mnemonic	Function
7:4	P0xIO	Unavailable on VMX1020
3	P0.3/RX1IE	0: General purpose input or UART1 RX When using **UART1 you must set this bit to 0.** 1: General purpose output
2	P0.2/TX1OE	0: General purpose input 1: General purpose output or or UART1 TX When using **UART1 you must set this bit to 1.**
1	P0.1/T2EXOE	0: General purpose input or Timer 2 EX When using **Timer 2 you must set this bit to 0.** 1: General purpose output
0	P0.0/T2INOE	0: General purpose input or Timer 2 In When using **Timer 2 you must set this bit to 0.** 1: General purpose output

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

TABLE 17: (P1PINCFG) PORT 1 PORT CONFIGURATION REGISTER - SFR 9CH

7	6	5	4
P1.7	P1.6	P1.5	P1.4

3	2	1	0
P1.3/PWM3	P1.2/PWM2	P1.1/PWM1	P1.0/PWM0

Bit	Mnemonic	Function
7	P1.7	0: General purpose input 1: General purpose output
6	P1.6	0: General purpose input 1: General purpose output
5	P1.5	0: General purpose input 1: General purpose output
4	P1.4	0: General purpose input 1: General purpose output
3	P1.3/PWM3OE	0: General purpose input 1: General purpose output or PWM bit 3 output When using **PWM you must set this bit to 1.**
2	P1.2/PWM2OE	0: General purpose input 1: General purpose output or PWM bit 2 output When using **PWM you must set this bit to 1**
1	P1.1/PWM1OE	0: General purpose input 1: General purpose output or PWM bit 1 output When using **PWM you must set this bit to 1**
0	P1.0/PWM0OE	0: General purpose input 1: General purpose output or PWM bit 0 output When using **PWM you must set this bit to 1**

Special Note about Port1

Port 1.0-P1.3 can be used as standard digital outputs. However, in order to do this, the Timer 2 clock must be enabled by setting the T2CLKEN bit of the DIGPWREN register. In addition, the Timer 2 register must also have the same reset value.

TABLE 18: (P2PINCFG) PORT 2 PORT CONFIGURATION REGISTER - SFR 9DH

7	6	5	4
P2.7/SDIE	P2.6/SDOE	P2.5/SCKIE	P2.4/SSIE

3	2	1	0
CS0OE	CS1OE	CS2OE	CS3OE

Bit	Mnemonic	Function
7	P2.7/SDIE	0: General purpose input or SDI When using the **SPI you must set this bit to 0.** 1: General purpose output
6	P2.6/SDOE	0: General purpose input 1: General purpose output or SDO When using the **SPI you must set this bit to 1.**
5	P2.5/SCKIE	0: General purpose input or SCK When using the **SPI you must set this bit to 0.** 1: General purpose output
4	P2.4/SSIE	0: General purpose input or Slave Select When using the **SPI you must set this bit to 0.** 1: General purpose output
3	P2.3/CS0OE	0: General purpose input 1: General purpose output or Chip Select bit 0 output When using the **SPI you must set this bit to 1.**
2	P2.2/CS1OE	0: General purpose input 1: General purpose output or Chip Select bit 1 output When using the **SPI you must set this bit to 1.**
1	P2.1/CS2OE	0: General purpose input 1: General purpose output or Chip Select bit 2 output When using **SPI you must set this bit to 1.**
0	P2.0/CS3OE	0: General purpose input 1: General purpose output or Chip Select bit 3 output When using **SPI you must set this bit to 3.**

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

TABLE 19: (P3PINCFG) PORT 3 PORT CONFIGURATION REGISTER - SFR 9EH

7	6	5	4
P3.7/MSCLOE	P3.6/MSDAOE	P3.5/T1INE	P3.4/CCU1IE

3	2	1	0
P3.3/CCU0IE	P3.2/T0INIE	P3.1/RX0IE	P3.0/TX0IE

Bit	Mnemonic	Function
7	P3.7/MSCLOE	0: General purpose input 1: General purpose output or Master I2C SCL output When using the **I2C you must set this bit to 1.**
6	P3.6/MSDAOE	0: General purpose input 1: General purpose output or Master I2C SDA When using the **I2C you must set this bit to 1.**
5	P3.5/T1INE	0: General purpose input or Timer1 Input When using Timer 1 **you must set this bit to 0.** 1: General purpose output
4	P3.4/CCU1IE	0: General purpose input or CCU1 Input When using the **Compare and Capture unit you must set this bit to 0.** 1: General purpose output
3	P3.3/CCU0IE	0: General purpose input or CCU0 Input When using the **Compare and Capture unit you must set this bit to 0.** 1: General purpose output
2	P3.2/T0INIE	0: General purpose input or Timer 0 Input When using Timer 0 **you must set this bit to 0.** 1: General purpose output
1	P3.1/RX0IE	0: General purpose input or UART0 Rx When using **UART0 you must set this bit to 0.** 1: General purpose output
0	P3.0/TX0IE	0: General purpose input 1: General purpose output or UART0 Tx When using **UART0 you must set this bit to 1.**

I/O usage example

The following example demonstrates the configuration of the VERSA MIX I/Os.

```
//-------------------------------------------------------------------------
//This example toggles all the IOs, first P0, then P1, P2 and P3.
//-------------------------------------------------------------------------
#pragma PATHINCLUDE(../include)
#pragma LARGE
#pragma INTVECTOR(0x00)
#pragma INTERVAL(0x08)
#pragma UNSIGNEDCHAR

#include <VMIXReg.h>

void waitABit()
{
  char i, max=50;
  for(i=0; i<max; i++) {};
}

at 0x0000 void main (void)
          {
  DIGPWREN = 0x80;              // Enable Timer 2
  P0PINCFG = 0xFF;              // Configure all Ports in output
   P1PINCFG = 0xFF;
  P2PINCFG = 0xFF;
  P3PINCFG = 0xFF;

  while(1)
   {
      P0 = ~P0;
      P1 = ~P1;
      P2 = ~P2;
      P3 = ~P3;
      waitABit();
    }
}
```


1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

MAC - Multiply Accumulator Unit

MAC Features

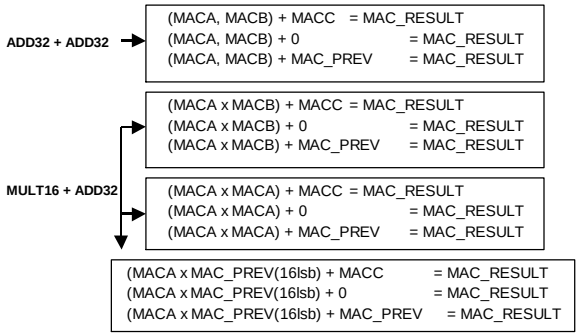
The VERSA MIX includes a hardware based multiply-accumulator unit, which is intended to provide a important speed up of arithmetic operations.

MAC unit facts list:

- Hardware Calculation Engine
- Calculation result is ready as soon as the input registers are loaded
- Signed mathematical calculations
- Unsigned MATH operations are possible if the MUL engine operands are limited to 15 bits in size
- Auto/Manual reload of MAC_RES
- Enhanced VERSA MIX MAC Unit
- Easy implementation of complex MATH operations
- 16 bit and 32-bit Overflow Flag
- 32-bit Overflow can raise an interrupt to the ?P
- MAC operand registers can be cleared individually or all together
- Overflow flags can be configured to stay active until manually cleared
- Can store and use results from previous operations

The MAC can be configured to perform the following operations:

FIGURE 8: VERSA MIX MAC OPERATION



Where MACA (multiplier), MACB (multiplicand), MACACC (accumulator) and MACRESULT (result) are 16, 16, 32 and 32 bits, respectively.

MAC Control Registers

At the exception of the Barrel Shifter, the MAC unit operation is controlled by two SFR registers: These registers are

- The MACCTRL
- The MACTRLC2

The following two tables describe the details of these control registers.

TABLE 20: (MACCTRL) MAC UNIT CONTROL REGISTER - SFR EBH

7	6	5	4
LOADPREV	PREVMODE	OVMODE	OVRDVAL

3	2	1	0
ADDSRC [1:0]		MULCMD [1:0]	

Bit	Mnemonic	Function
7	LOADPREV	MACPREV manual Load control 1 = Manual load of the MACPREV register content if PREVMODE = 1
6	PREVMODE	Loading method of MACPREV register 0 = Automatic load when MACA0 is written. 1 = Manual Load when 1 is written into LOADPREV
5	OVMODE	0 = Once set by math operation, the OV16 and OV32 flag will remain set until the overflow condition is removed. 1= Once set by math operation, the OV16 and OV32 flag will stay set until it is cleared manually.
4	OVRDVAL	0 = The value on MACRES is the calculation result. 1 = the value on MACRES is the 32LSB of the MACRES when the OV32 overflow occurred
3:2	ADDSRC	32-bit Addition source **B Input** 00 = 0 (No Add) 01 = C (std 32-bit reg) 10 = RES –1 11 = C **A Input** 00=Multiplication 01=Multiplication 10=Multiplication 11= Concatenation of {A,B} for 32 bit addition
1:0	MULCMD	Multiplication Command 00 = MACA x MACB 01 = MACA x MACA 10 = MACA x MACPREV (16 LSB) 11 = MACA x MACBB

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

TABLE 21: (MACCTRL2) MAC UNIT CONTROL REGISTER 2 -SFR F1H

7	6	5	4
MACCLR2 [2:0]			MACOV32IE

3	2	1	0
-	-	MACOV16	MACOV32

Bit	Mnemonic	Function
7:5	MACCLR	MAC Register Clear 000 = No Clear 001 = Clear MACA 010 = Clear MACB 011 = Clear MACC 100 = Clear MACPREV 101 = Clear All MAC regs + Overflow Flags 110 = Clear Overflow Flags only
4	MACOV32IE	MAC 32-bit Overflow IRQ Enable
3	-	-
2	-	-
1	MACOV16	16-bit Overflow Flag 0 = No 16 overflow 1 = 16-bit MAC Overflow occurred
0	MACOV32	32-bit Overflow Flag 1 = 32-bit MAC Overflow This automatically loads the MAC32OV register. The MACOV32 can generate a MAC interrupt when enabled.

MAC Unit Data Registers

The MAC Data registers include operand and result registers that serve to store the numbers being manipulated in mathematical operations. Some of these registers are uniquely for addition (such as MACC) while others can be used for all operations. The MAC operation registers are represented below.

MACA and MACB Multiplication (Addition) input registers

The MACA and MACB register serves as the 16bit input operands when performing multiplication.

When the MAC is configured to perform 32bit addition, the MACA and the MAC B registers are concatenated to perform a 32bit word. In that case the MACA register contains the upper 16bit of the 32bit operand and the MACB contains the lower 16bit

TABLE 22: (MACA0) MAC UNIT A OPERAND, LOW BYTE - SFR F2H

7	6	5	4	3	2	1	0
MACA0 [7:0]							

Bit	Mnemonic	Function
7:0	MACA0	Lower segment of the MACA operand

TABLE 23: (MACA1) MAC UNIT A OPERAND, HIGH BYTE - SFR F3H

7	6	5	4	3	2	1	0
MACA1 [15:8]							

Bit	Mnemonic	Function
15:8	MACA1	Upper segment of the MACA operand

TABLE 24: (MACB0) MAC UNIT B OPERAND, LOW BYTE - SFR F9H

7	6	5	4	3	2	1	0
MACB0 [7:0]							

Bit	Mnemonic	Function
7:0	MACB0	Lower segment of the MACB operand

TABLE 25: (MACB1) MAC UNIT B OPERAND, HIGH BYTE - SFR FAH

7	6	5	4	3	2	1	0
MACB1 [7:0]							

Bit	Mnemonic	Function
7:0	MACB1	Upper segment of the MACB operand

MACC input register

The MACC register is a 32bit register used to perform 32bit addition.

It is possible to substitute the MACPREV Register to the MAC C register or 0 in the 32bit addition.

TABLE 26: (MACC0) MAC UNIT C OPERAND, LOW BYTE - SFR ECH

7	6	5	4	3	2	1	0
MACC0 [7:0]							

Bit	Mnemonic	Function
7:0	MACC0	Lower segment of the 32-bit addition register

TABLE 27: (MACC1) MAC UNIT C OPERAND, BYTE 1 - SFR EDH

7	6	5	4	3	2	1	0
MACC1 [15:8]							

Bit	Mnemonic	Function
15:8	MACC1	Lower middle segment of the 32-bit addition register

TABLE 28: (MACC2) MAC UNIT C OPERAND, BYTE 2 - SFR EEH

7	6	5	4	3	2	1	0
MACC2 [23:16]							

Bit	Mnemonic	Function
23:16	MACC2	Upper middle segment of the 32-bit addition register

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

TABLE 29: (MACC3) MAC UNIT C OPERAND, HIGH BYTE - SFR EFH

7	6	5	4	3	2	1	0
MACC3 [31:24]							

Bit	Mnemonic	Function
31:24	MACC3	Upper segment of the 32-bit addition register

MACRES Result register

The MACRES register, which is 32bit wide, contains the result of the MAC operation. In fact, the MACRES register is the output of the Barrel Shifter.

TABLE 30: (MACRES0) MAC UNIT RESULT, LOW BYTE - SFR F4H

7	6	5	4	3	2	1	0
MACRES0 [7:0]							

Bit	Mnemonic	Function
7:0	MACRES0	Lower segment of the 32-bit MAC result register

TABLE 31: (MACRES1) MAC UNIT RESULT, BYTE 1 - SFR F5H

7	6	5	4	3	2	1	0
MACRES1 [15:8]							

Bit	Mnemonic	Function
15:8	MACRES1	Lower middle segment of the 32-bit MAC result register

TABLE 32: (MACRES2) MAC UNIT RESULT, BYTE 2 - SFR F6H

7	6	5	4	3	2	1	0
MACRES2 [23:16]							

Bit	Mnemonic	Function
23:16	MACRES2	Upper middle segment of the 32-bit MAC result register

TABLE 33: (MACRES3) MAC UNIT RESULT, HIGH BYTE - SFR F7H

7	6	5	4	3	2	1	0
MACRES3 [31:24]							

Bit	Mnemonic	Function
31:24	MACRES3	Upper segment of the 32-bit MAC result register

MACPREV register

The MACPREV register provides the ability to automatically or manually save the content of the MACRES register and re-inject it into the calculation. This feature is especially useful in applications where the result of a given operation serves as one of the operand of the next one.

As it has been mentioned earlier, there are two ways, to load the MACPREV register controlled by the PREVMODE bit value:

PREVMODE = 0:

Auto MACPREV load by writing into the MACA0 register. Selected when PREVMODE = 0.

PREVMODE = 1:

Manual load of MACPREV when the LOADPREV bit is set to 1

A good example using the auto loading of the MACPREV feature is the implementation of and FIR Filter. In that specific case, it is possible to save a total of 8 MOV operations per node calculation.

TABLE 34: (MACPREV0) MAC UNIT PREVIOUS OPERATION RESULT, LOW BYTE - SFR FCH

7	6	5	4	3	2	1	0
MACPREV0 [7:0]							

Bit	Mnemonic	Function
7:0	MACPREV0	Lower segment of 32-bit MAC previous result register

TABLE 35: (MACPREV1) MAC UNIT PREVIOUS OPERATION RESULT, BYTE 1 - SFR FDH

7	6	5	4	3	2	1	0
MACPREV1 [7:0]							

Bit	Mnemonic	Function
15:8	MACPREV1	Lower middle segment of 32-bit MAC previous result register

TABLE 36: (MACPREV2) MAC UNIT PREVIOUS OPERATION RESULT, BYTE 2 - SFR FEH

7	6	5	4	3	2	1	0
MACPREV2 [15:8]							

Bit	Mnemonic	Function
23:16	MACPREV2	Upper middle segment of 32-bit MAC previous result register

TABLE 37: (MACPREV3) MAC UNIT PREVIOUS OPERATION RESULT, HIGH BYTE - SFR FFH

7	6	5	4	3	2	1	0
MACPREV3 [7:0]							

Bit	Mnemonic	Function
31:24	MACPREV3	Upper segment of 32-bit MAC previous result register

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

MAC Barrel Shifter

The MAC includes a 32-bit Barrel Shifter at the output of the 32-bit addition unit. The Barrel Shifter can perform right/left shift operations which is useful to scale the output result of the MAC.

The shifting range is adjustable from 0 to 16 both ways. The “shifted” addition unit output can be routed to:

- MAC_RES
- MAC_PREV
- MAC_OV32

The barrel shifter can perform both arithmetic and logical shifts: The shift left operation can be configured as an arithmetic or logical shift. In the later, the sign bit is discarded.

TABLE 38: (MACSHIFTCTRL) MAC UNIT BARREL SHIFTER CONTROL REGISTER - SFR FBH

7	6	5	4	3	2	1	0
SHIFTMODE	ALSHSTYLE	SHIFTAMPL [5:0]					

Bit	Mnemonic	Function
7	SHIFTMODE	0 = Logical SHIFT 1 = Arithmetic SHIFT
6	ALSHSTYLE	Arithmetic Shift Left Style 0= Arithmetic Left Shift: Logical Left 1= Arithmetic Left Shift: Keep sign bit
5:0	SHIFTAMPL	Shift Amplitude 0 to 16 (5 bits to provide 16 bits) Negative Number = Shift Right 2 complements Positive Number = Shift Left

Setting up the MAC Unit

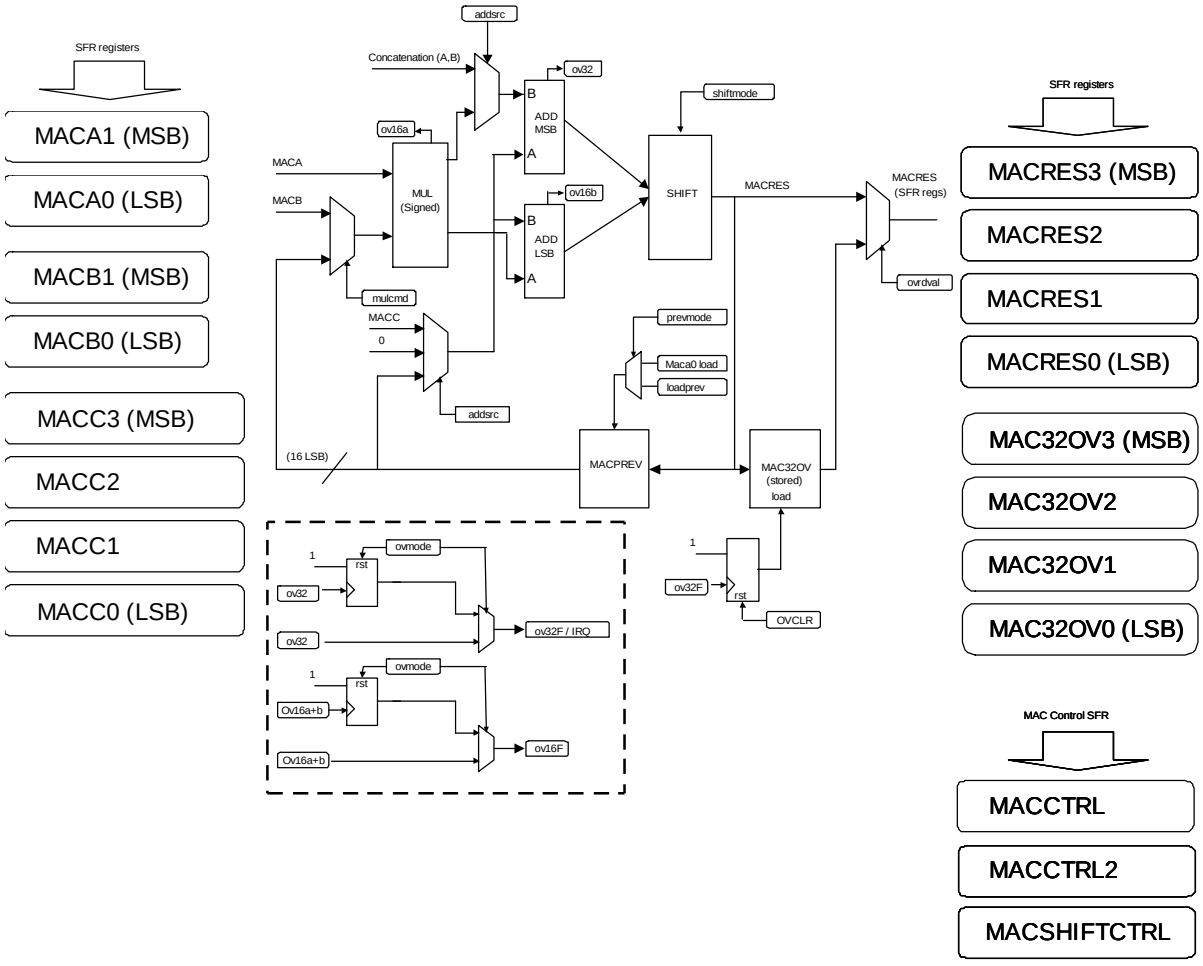
In order use the MAC unit, one must first set up and configure the module. The following lines of code can be used to perform these tasks. The first part of the code is the interrupt setup and module configuration, whereas the second part is the interrupt function.

Sample C code for MAC Unit interrupt setup and module configuration:

```
//-------------------------------------------------------------------------

// Sample C code to setup the MAC unit
//-------------------------------------------------------------------------
// MAC setup

IEN0 |= 0x80;                   // Enable all interrupts
IEN1 |= 0x10;                   // Enable MAC interrupt
DIGPWREN |= 0x20;               // Enable MAC unit
MACCTRL1 = 0x0C;                // {A,B}+C
MACCTRL2 = 0x10;                // Enable INT overflow_32

// MAC example use

MACA0 = 0xFF;
MACA1 = 0x7F;
MACB0 = 0xFF;
MACB1 = 0xFF;
MACC0 = 0xFF;
MACC1 = 0xFF;
MACC2 = 0xFF;
MACC3 = 0x7F;

//-------------------------------------------------------------------------
// Interrupt Function

void int_5_mac (void) interrupt 12
{
IEN0 &= 0x7F;                   // Disable all interrupts

        /*-----------------------*/
        /*Interrupt code here*/
        /*-----------------------*/

IRCON &= 0xEF;                  // Clear flag (IEX5)
IEN0 |= 0x80;                   // Enable all interrupts
}
//-------------------------------------------------------------------------
```


1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

FIGURE 9: VERSA MIX MAC FUNCTIONAL DIAGRAM



The block diagram above shows the interaction between the registers and the other components that comprise the MAC unit on the VERSA MIX.

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Timers/Counters

The VERSA MIX includes 3 general-purpose timer/counters (Timer 0, Timer 1, and Timer 2), two 10-bit baud rate generators reserved for UARTS. Timer 0 and Timer 1 can operate as either a timer with a clock rate based on the system clock, or as an event counter clocked by the T0IN (Timer 0) and T1IN (Timer 1).

Timer 2 can only operate in 16-bit timer mode. This timer can also be used as a serial port baud rate generator.
Each general-purpose timer/counter consists of a 16-bit register that is accessible by software as two independent SFR registers.

- **Timer 0** -TL0 and TH0
- **Timer 1** -TL1 and TH1
- **Timer 2** - TL2 and TH2

TABLE 39: (TL0) TIMER 0 LOW BYTE - SFR 8AH

7	6	5	4	3	2	1	0
TL0 [7:0]							

TABLE 40: (TL1) TIMER 1 LOW BYTE - SFR 8BH

7	6	5	4	3	2	1	0
TL1 [7:0]							

TABLE 41: (TH0) TIMER 0 HIGH BYTE - SFR 8CH

7	6	5	4	3	2	1	0
TH0 [7:0]							

TABLE 42: (TH1) TIMER 1 HIGH BYTE - SFR 8DH

7	6	5	4	3	2	1	0
TH1 [7:0]							

Timers 0 and 1

Timers 0 and 1 can be configured to operate in one of the following modes:

- Mode 0: 13-bit timer/counter
- Mode 1: 16-bit timer/counter
- Mode 2: 8-bit counter with auto-reload
- Mode 3: Two 8-bit counters

The configuration and control of the Timer 0 and Timer 1 is performed via the TMOD and the TCON SFR registers
The table below shows the TCON special function register of the VERSA MIX. This register contains the Timer 0/1 overflow flags, the timer 0/1 run control bits; the interrupt 0/1 edge flags; and the interrupt 0/1 interrupt type control bits.

TABLE 43: (TCON) TIMER 0, TIMER 1 TIMER/COUNTER CONTROL - SFR 88H

7	6	5	4
TF1	TR1	TF0	TR0

3	2	1	0
IE1	IT1	IE0	IT0

Bit	Mnemonic	Function
7	TF1	Timer 1 overflow flag set by hardware when Timer 1 overflows. This flag can be cleared by software and is automatically cleared when interrupt is processed.
6	TR1	Timer 1 Run control bit. If cleared Timer 1 stops.
5	TF0	Timer 0 overflows flag set by hardware when Timer 0 overflows. This flag can be cleared by software and is automatically cleared when interrupt is processed.
4	TR0	Timer 0 Run control bit. If cleared timer 0 stops.
3	IE1	Interrupt 1 edge flag. Set by hardware when falling edge on external INT1 is observed. Cleared when interrupt is processed.
2	IT1	Interrupt 1 type control bit. Selects falling edge or low level on input pin to cause interrupt.
1	IE0	Interrupt 0 edge flag. Set by hardware when falling edge on external pin INT1 is observed. Cleared when interrupt is processed.
0	IT0	Interrupt 0 type control bit. Selects falling edge or low level on input pin to cause interrupt.

The TMOD register is mainly used to set the operating mode of the timers and it allows the user to enable the external gate control and to select a timer or counter operation.

TABLE 44: (TMOD) TIMER MODE CONTROL - SFR 89H

7	6	5	4
GATE1	CT1	M11	M01

3	2	1	0
GATE0	CT0	M10	M00

Bit	Mnemonic	Function
7	GATE1	If set, enables external gate control (pin INT1 for Counter 1). When INT1 is high, and TRx bit is set (see TCON register), a counter is incremented every falling edge on the T1IN input pin.
6	CT1	Selects timer or counter operation. 1 = A counter operation is performed 0 = The corresponding register will function as a timer.
5	M11	Selects mode for Timer/Counter 1, as shown in Table 39.
4	M01	Selects mode for Timer/Counter 1, as shown in Table 39.
3	GATE0	If set, enables external gate control (pin INT0 for Counter 0). When INT0 is high, and TRx bit is set (see TCON register), a counter is incremented every falling edge on the T0IN input pin.
2	CT0	Selects timer or counter operation. 1 = A counter operation is performed 0 = The corresponding register will function as a timer.
1	M10	Selects mode for Timer/Counter 0, as shown in Table 45.
0	M00	Selects mode for Timer/Counter 0, as shown in Table 45.

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

The operating mode of the Timer 0 and the Timer 1 is determined by the value of the M1x and M0x bits of the TMOD register.

The table below summarizes the four modes of operation of Timers 0 and 1.

TABLE 45: TIMER/COUNTER MODE DESCRIPTION SUMMARY

M1	M0	Mode	Function
0	0	Mode 0	13-bit Counter/Timer, with 5 lower bits in TL0 or TL1 register and bits in TH0 or TH1 register (for timer 0 and timer, respectively). The 3 high order bits of TL0 and TL1 are held at 0.
0	1	Mode 1	16-bit Counter/Timer
1	0	Mode 2	8-bit auto-reload Counter/Timer. The reload value is kept in TH0 or TH1, while TL0 or TL1 is incremented every machine cycle. When TLx overflows, a value from THx is copied to TLx.
1	1	Mode 3	If Timer 1 M1 and M0 bits are set to 1, Timer 1 stops. If Timer 0 M1 and M0 bits are set to 1, Timer 0 acts as two independent 8-bit Timers/Counters.

FIGURE 10: TIMER 0 MODE 0

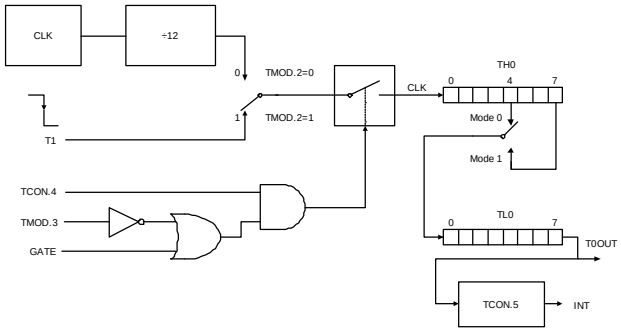
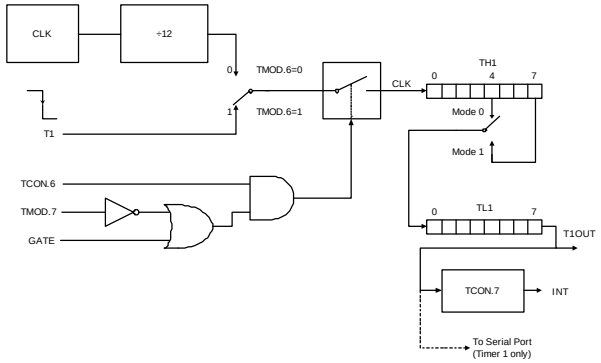


FIGURE 11: TIMER 1 MODE 0



Mode 0 (13-bit)

Mode 0 operation is the same for Timer 0 and Timer 1. In Mode 0, the timer is configured as a 13-bit counter that uses bits 0-4 of TL0 (or TL 1) and all 8 bits of TH0 (or TH1). The timer enable bit (TR0/TR1) in the TCON SFR (table 37) starts the timer. The CT bit selects the Timer/Counter clock source, CLK or T0IN/T1IN.

When the 13-bit count increments from 1FFFh (all ones), the counter rolls over to all zeros, the TF0 (or TF1) bit is set in the TCON SFR, and the T0OUT (or T1OUT) pin goes high for one clock cycle.

The upper 3 bits of TL0 (or TL1) are indeterminate in Mode 0 and must be masked when the software evaluates the register.

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Mode1 (16-bit)

Mode 1 operation is the same for Timer 0 and Timer 1. In Mode 1, the timer is configured as a 16-bit counter. The counter rolls over to all zeros (0000h) upon surpassing FFFFh. Otherwise; Mode 1 operation is the same as Mode 0.

FIGURE 12: TIMER 0 MODE 1

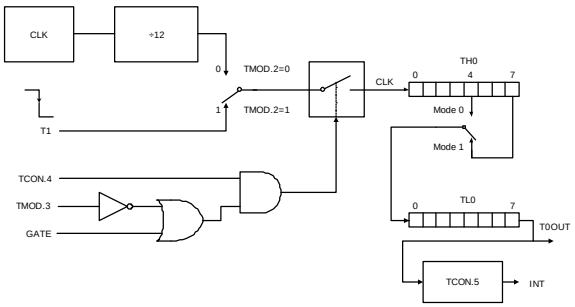
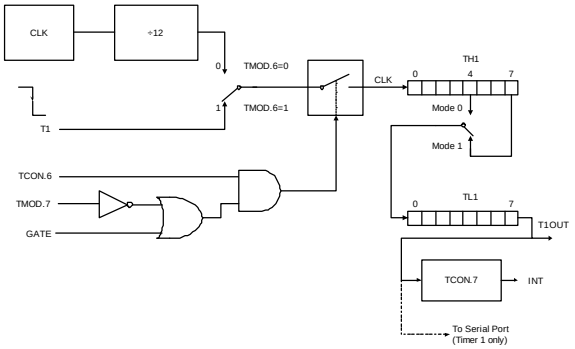


FIGURE 13: TIMER 1 MODE 1



Mode 2 (8-bit)

The operation of Mode 2 is the same for Timer 0 and Timer 1. In Mode 2, the timer is configured as an 8-bit counter, with automatic reload of the start value. The LSB register (TL0 or TL1) is the counter and the MSB register (TH0 or TH1) stores the reload value.

The Mode 2 counter control is the same as for Mode 0 and Mode 1. However, in Mode 2, when TLx surpasses FFh, the value stored in THx is reloaded into TLx.

FIGURE 14: TIMER 0 MODE 2

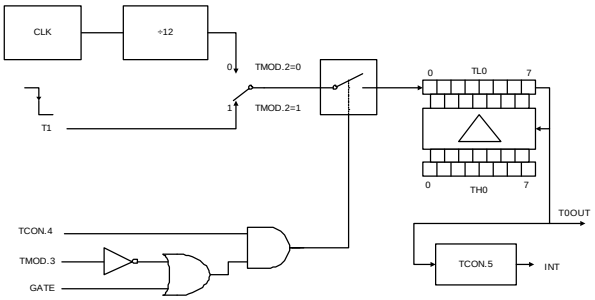
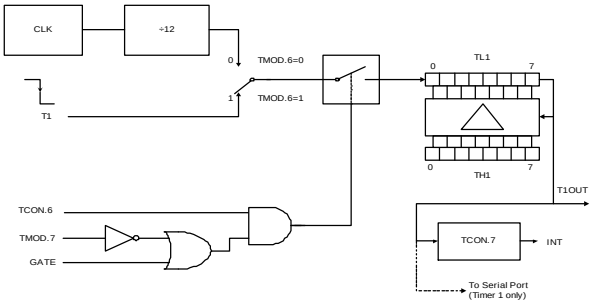


FIGURE 15: TIMER 1 MODE 2

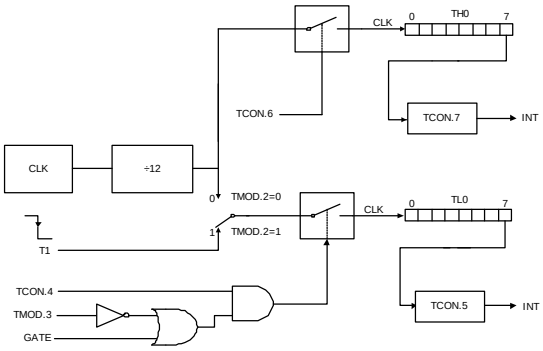


1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Mode 3 (2 x 8-bit)

In Mode 3, Timer 0 operates as two 8-bit counters and Timer 1 stops counting and holds its value.

FIGURE 16: TIMER MODE 3



Setting Up Timer 0

In order to use Timer 0, one must first set up and configure the module. The following lines of code can be used to perform these tasks. The first part of the code is the interrupt setup and module configuration whereas the second part is the interrupt function.

Sample C code to set up Timer 0:

```
//-------------------------------------------------------------------------
// Sample C code to setup Timer 0
//-------------------------------------------------------------------------

// TIMER 0 SETUP

IEN0 |= 0x80;                    // ENABLE ALL INTERRUPTS
IEN0 |= 0x02;                    // ENABLE INTERRUPT TIMER 0
TCON = 0x10;                     // ENABLE TIMER 0
TMOD = 0x02;                     // TIMER 0 MODE 2

// TIMER 0 EXAMPLE USE
TL0 = 0xE0;                      // SET TIMER 0 OFFSET

//-------------------------------------------------------------------
// INTERRUPT FUNCTION

VOID INT_TIMER_0 (VOID) INTERRUPT 1
{
IEN0 &= 0x7F;                    // DISABLE ALL INTERRUPTS

        /*-------------------------*/
        /*Interrupt code here*/
        /*-------------------------*/

IEN0 |= 0x80;                    // Enable all interrupts
}
//-------------------------------------------------------------------------
```

Setting Up Timer 1

The following is the code used to configure Timer 1. The first part of the code is the interrupt setup and module configuration whereas the second part is the interrupt function.

Sample C code to set up Timer 1

```
//-------------------------------------------------------------------------
// Sample C code to setup TIMER1
//-------------------------------------------------------------------------


// TIMER 1 setup


IEN0 |= 0x80;                    // Enable all interrupts
IEN0 |= 0x08;                    // Enable interrupt Timer1
TMOD = 0x20;                     // Timer 1 mode 2
TCON = 0x40;                     // Enable Timer 1
                                 // TIMER1 example use
TL1 = 0xFC;                      // Timer1 offset


//-------------------------------------------------------------------

// Interrupt function

void int_timer_1 (void) interrupt 3
{
IEN0 &= 0x7F;                    // Disable all interrupts

        /*-------------------------*/
        /*Interrupt code here*/
        /*-------------------------*/


IEN0 |= 0x80;                    // Enable all interrupts
}

//-------------------------------------------------------------------------
```


1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Timer 2

The VERSA MIX Timer 2 provides the following functionalities:

- 16-Bit Timer
- 16-Bit Auto-Reload Timer
- 16 Bit Precision Baud Rate Generator

Dividing the Clock Frequency

In the timer modes, the clocking of Timer 2 can only be accomplished using distinct clock division values. These values are $f_{clk}/2$, $f_{clk}/12$ and $f_{clk}/24$.

The Timer 2 clocking frequency is defined by the value of the T2PS (Timer 2 prescaler) and T2PSM (Timer 2 master prescaler) bits of the T2CON register as shown in Table 39 (T2CON).

The figure 17 show the Timer2 / Compare/ capture unit block diagram.

Pulse Width Modulation (PWM)

The VERSA MIX includes four PWM outputs shared with P1.0 to P1.3 I/O pins.

There are multiple applications for PWM such as:

- D/A conversion
- Motor control
- Light control
- Etc.

The clock source of the PWM is derived from the Timer 2.which is incremented at every signal pulse of the appropriate source. This source is selected by the T2I1 and T2I0 bits of T2CON register.

The PWM signal generation is derived from the comparison result between the values stored into the capture compare registers and the Timer2 value.

When a digital value is written into one of the CCx or CRCx registers, a comparison occurs (providing that Timer 2 is in compare mode of course). As long as the value of the CCxx or CRCx register is greater than the Timer 2 value, the compare unit will output a logical low.

When the value of Timer2 equals the value of the CCx or CRCx register, the compare unit will change from a logical low to a logical high.

Using the PWM as D/A converter

One of the uses of the PWM is to perform D/A conversion. It is possible to perform a digital to analog conversion by placing the modulated signal at the input of a low pass filter. The greater the duty cycle of the square wave, the greater the DC value is at the output of the low pass filter and vice versa. Below is a table that summarizes the relationship between the duty cycle and the output that it will generate on the VERSA MIX. (Note that the values are given with respect to the sinusoidal ripple that occurs at the output of the filter)

TABLE 46: PULSE WIDTH MODULATION

	RIPPLE (mV)			
Duty Cycle	**Max V (mV)**	**Min V (mV)**	**Delta V (mV)**	Stable (ms) ~
1% (T=0.16us)	81.89	80.6	1.271	4.5
10% (T= 1.6us)	817.54	807.41	10.125	5.8806
25% (T= 4us)	1288	1273	15.142	5.7982
50% (T=8us)	2541	25213	19.84	7.5483
75% (T=12us)	3788	3773	14.65	6.6
100% (T=16us)	5	5	0	4.988

As explained above, a variation in width allows us to generate various DC values, which is equivalent to performing a digital to analog conversion.

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Selecting the Data Line Width

The VERSA MIX is capable of comparing and capturing data using data lines up to 16 bits wide. When comparing 2 registers or capturing 1 register, it is required to set the T2S bit of the T2CON register to 1. This adjusts the lines to have an 8-bit width.

On the other hand, when comparing two pairs of registers, for example CCH1 and CCL1 to TH2 and TL2, the T2SIZE bit must be set to 0. This will make the data lines 16 bits wide.

TABLE 47: (T2CON) TIMER 2 CONTROL REGISTER -SFR C8H

7	6	5	4
T2PS	T2PSM	T2SIZE	T2RM1

3	2	1	0
T2RM0	T2CM	T2IN1	T2IN0

Bit	Mnemonic	Function
7	T2PS	Prescaler select bit: 0 = Timer 2 is clocked with 1/12 of the oscillatory frequency 1 = Timer 2 is clocked with 1/24 of the oscillatory frequency
6	T2PSM	0 = Prescaler 1 = clock/2
5	T2SIZE	Timer 2 Size 0 = 16-bit 1 = 8-bit
4	T2RM1	Timer 2 reload mode selection
3	T2RM0	0X = Reload disabled 10 = Mode 0 11 = Mode 1
2	T2CM	Timer 2 compare mode selection 0 = Mode 0 1 = Mode 1
1	T21IN1	Timer 2 input selection
0	T21IN0	00 = Timer 2 stops 01 = Input frequency f/2, f/12 or f/24 10 = Timer 2 is incremented by external signal at pin T2 (P1.7) 11 = Internal clock input is gated to the Timer 2.

Compare, Capture and Reload Registers

The following tables represent the different registers that may capture or that may be compared to the value of Timer 2 (there are 8 in total). Please note that the CRCx registers are the only ones that can be used to perform a reload operation.

TABLE 48: (CCL1) COMPARE/CAPTURE REGISTER 1, LOW BYTE - SFR C2H

7	6	5	4	3	2	1	0
CCL1 [7:0]							

TABLE 49: (CCH1) COMPARE/CAPTURE REGISTER 1, HIGH BYTE - SFR C3H

7	6	5	4	3	2	1	0
CCH1 [7:0]							

TABLE 50: (CCL2) COMPARE/CAPTURE REGISTER 2, LOW BYTE - SFR C4H

7	6	5	4	3	2	1	0
CCL2 [7:0]							

TABLE 51: (CCH2) COMPARE/CAPTURE REGISTER 2, HIGH BYTE - SFR C5H

7	6	5	4	3	2	1	0
CCH2 [7:0]							

TABLE 52: (CCL3) COMPARE/CAPTURE REGISTER 3, LOW BYTE - SFR C6H

7	6	5	4	3	2	1	0
CCL3 [7:0]							

TABLE 53: (CCH3) COMPARE/CAPTURE REGISTER 3, HIGH BYTE - SFR C7H

7	6	5	4	3	2	1	0
CCH3 [7:0]							

TABLE 54: (CRCL) COMPARE/RELOAD/CAPTURE REGISTER, LOW BYTE - SFR CAH

7	6	5	4	3	2	1	0
CRCL [7:0]							

TABLE 55: (CRCH) COMPARE/RELOAD/CAPTURE REGISTER, HIGH BYTE - SFR CBH

7	6	5	4	3	2	1	0
CRCH [7:0]							

TABLE 56: (TL2) TIMER 2, LOW BYTE - SFR CCH

7	6	5	4	3	2	1	0
TL2 [7:0]							

TABLE 57: (TH2) TIMER 2, HIGH BYTE - SFR CDH

7	6	5	4	3	2	1	0
TH2 [7:0]							

Overflows and Interrupts

When the Timer 2 counter overflows from FFFFh, the TF2 flag is set, which can raises a Timer 2 interrupt (if enabled).

A Timer 2 interrupt may also be raised from T2EX if EXEN2 is set. An interrupt of this sort may also be verified by checking the value of the T2IF bit of the IRCON register.

The compare and capture function on the VERSA MIX has a wide variety of different operating states. These states are controlled by the COCALX and COCAHX registers. The different states are: compare and capture mode enable/disable; capture on falling edge at CC0 enable; and compare enabled. The detailed signal combinations are summarized in the following table.

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

TABLE 58: (CCEN) COMPARE/CAPTURE ENABLE REGISTER -SFR C9H

7	6	5	4
COCAH3	COCAL3	COCAH2	COCAL2

3	2	1	0
COCAH1	COCAL1	COCAH0	COCAL0

Bit		
Mnemonic	Mnemonic	Function
COCAH0	**COCAL0**	Compare and capture mode for CRC register
0	0	Compare/capture disabled
0	1	Capture on falling edge at pin CC0 (1 cycle)
1	0	Compare enabled
1	1	Capture on write operation into register CRC1
COCAH1	**COCAL1**	Compare/capture mode for CC register 1
0	0	Compare/capture disabled
0	1	Capture on rising edge at pin CC1 (2 cycles)
1	0	Compare enabled
1	1	Capture on write operation into register CCL1
COCAH2	**COCAL2**	Compare/capture mode for CC register 2
0	0	Compare/Capture disabled
0	1	Capture on rising edge at pin CC2 (2 cycles)
1	0	Compare enabled
1	1	Capture on write operation into register CCL2
COCAH3	**COCAL3**	Compare/Capture mode for CC register 3
0	0	Compare/capture disabled
0	1	Capture on rising edge (2 cycles)
1	0	Compare enabled
1	1	Capture on write operation into register CCL3

Timer 2 16-bit Timer Modes

Event Counter Mode

When operating in the Event Counter Mode, the timer is incremented as soon as the external signal T2IN changes value from 1 to 0. A sample of the T2IN input is taken at every machine cycle. Timer 2 is incremented in the cycle following the one in which the transition was detected.

Gated Timer Mode

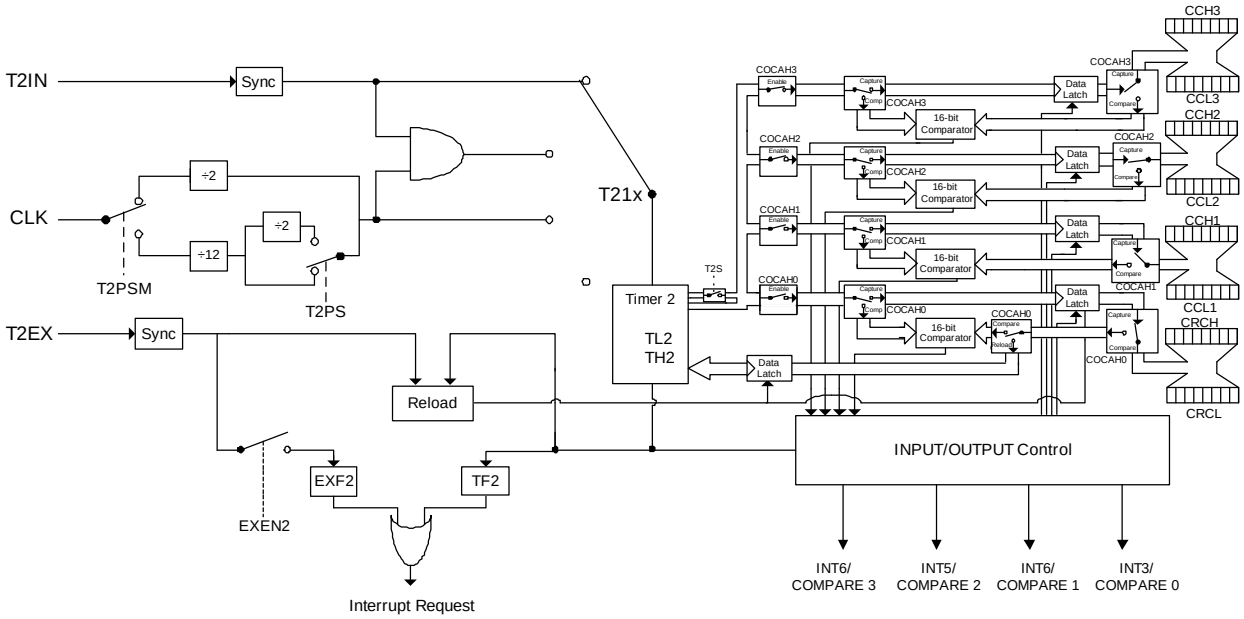
When operating in the Gated Timer Mode, the internal clock, which incremented timer 2, is gated by the external signal T2IN. In other words, when T2IN is high, the internal clock is allowed to pass through the AND gate. A low value of T2IN will stop the clock pulse, which allows for easier pulse width measurements.

The functionality and register interactions of the Timer 2 and the Compare, Capture and Reload Modes are depicted in figure 17.

Notice from the figure that the input selection of Timer 2 has 4 possible sources determined by 2 bits in the T2CON register. These select bits are named T211 and T210. The purpose of these bits is to select whether Timer 2 stops, takes its input from the f/12 or f/24 line, if it is incremented by an external signal at pin T2IN or if the internal clock input is gated to Timer 2.

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

FIGURE 17: TIMER 2 AND COMPARE/CAPTURE UNIT



The Timer 2 unit and its surrounding peripherals include a number of other functionalities explained in the following paragraphs:

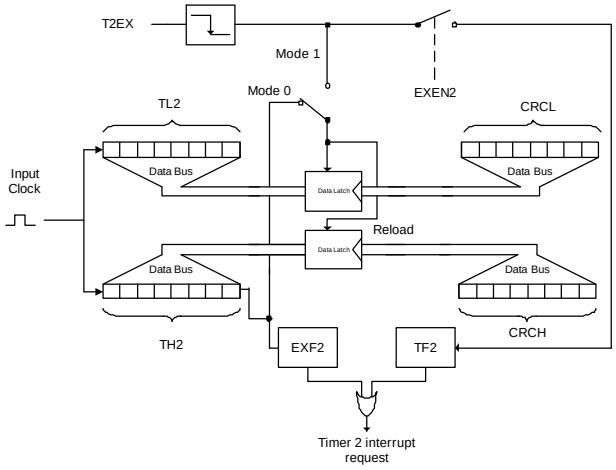
Reload Modes

Figure 18 (Timer 2 Reload mode) shows what happens during a reload operation. From the schematic, we see that CRCx constantly reloads TL2 and/or TH2 at each clock pulse. The reload mode is selected by the T2RM1 and T2RM0 bits of the T2CON register.

In Mode 0, when the timer overflows and all 1's become 0's, the TF2 overflow flag is set. Concurrently, this overflow causes the Timer 2 registers to be loaded with the 16-bit value contained in the CRCx register (which has been preset by software). This reload operation will occur during the same clock cycle in which TF2 was set, thus overwriting the present count value 0000h.

In Mode 1, a 16-bit reload from the CRCx register on the falling edge of T2EX occurs. This transition will set EXF2 if EXEN2 is set. This action will cause an interrupt (providing that the Timer 2 interrupt is enabled).

FIGURE 18: TIMER 2 RELOAD MODE



1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Compare Modes

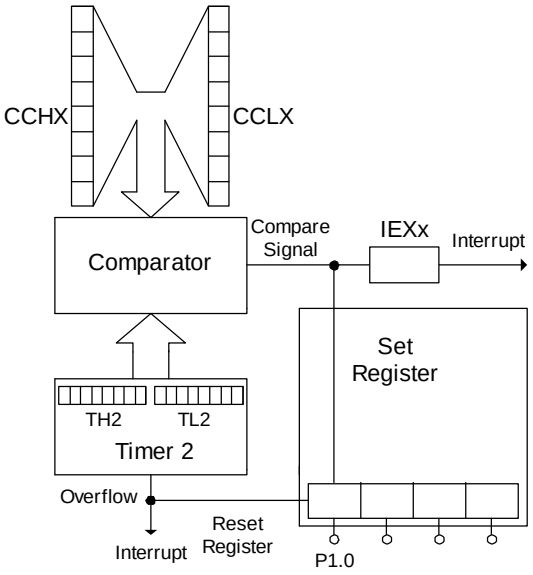
When Timer 2 is in compare mode, a Timer 2 count value is compared to a value that is stored in the CCxx or CRCx registers. If the values compared match (i.e. when the pulse changes state), an interrupt is generated at the appropriate port pin. Varying the value of the CCxx or CRCx allows a variation of the rectangular pulse generated at the output. This variation can be used to perform pulse width modulation.

In order to select the compare mode, we must adjust the value of bit T2CM in the T2CON register. In both modes the new value arrives at port pin 1 in the same clock cycle during which the internal compare signal is activated.

Compare Mode 0

The functional diagram of this mode can be found below in Figure 19. From this schematic, we notice that upon comparison of the 16-bit values of the CCxx and Txx (Timer 2 registers), a high compare signal is generated. The compare signal is then propagated to IEXx (which will generate an interrupt) and to the pin P1.0. Pin P1.0 is reset when a Timer 2 overflow occurs. It is important to note that when compare mode is enabled, the corresponding output pin can only be reset by the internal timer circuitry.

FIGURE 19: TIMER 2 COMPARE MODE 0 BLOCK DIAGRAM



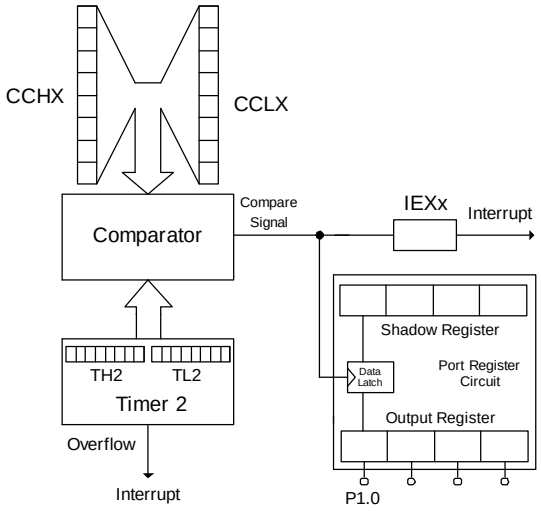
Compare Mode 1

This mode should be used when output signals are not related to a constant signal period.

If the unit is operating in Mode 1, and the software writes to the corresponding output register at the port, the new value will not appear at the output until the next compare match occurs.

One may therefore, decide whether the output signal is to make a new transition, or if it is to keep the old value that Timer 2 contained when it last matched the values in the CCxx or CRCx registers.

FIGURE 20: TIMER 2 COMPARE MODE 1 BLOCK DIAGRAM



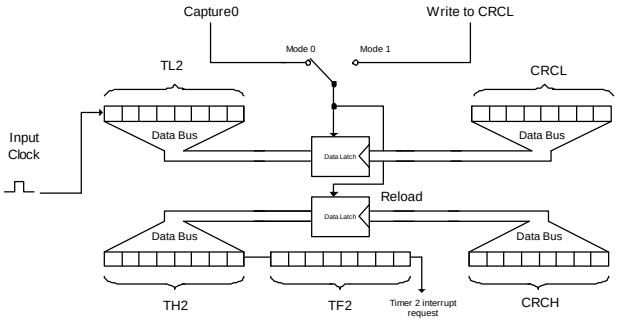
1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Capture Modes

In capture mode, the CCxx or CRCx latches the data placed on the data lines by Timer 2. In Mode 0, an external event triggers the latching of Timer 2's data by one of the compare and capture registers. In Mode 1, a capture will occur upon writing to the Low Byte of the chosen capture register. This mode allows the software to continuously read the contents of Timer 2.

Note that on VMX1020, the CCU3 input is not available.

FIGURE 21: TIMER 2 CAPTURE MODE 0 FOR CRCL AND CRCH BLOCK DIAGRAM



Setting up Timer 2

In order to use Timer 2, one must first set up and configure the module. The following lines of code may be used to perform these tasks.

```
//---------------------------------------------------------------------------
// Sample C code to setup Timer 2
//---------------------------------------------------------------------------

// TIMER_2 setup

IEN0 |= 0x80;                  // Enable all interrupts
IEN0 |= 0x20;                  // Enable interrupt Timer 2
DIGPWREN = 0x80;               // Disable module UART 1,
                               // and enable Timer 2
T2CON = 0x01;                  // Set timer 2 to OSC/12
TL2 = 0xE0;
TH2 = 0xFF;

//---------------------------------------------------------------------------

// Interrupt Function

void int_timer_2 (void) interrupt 5
{
IEN0 &= 0x7F;                  // Disable all interrupts

        /*-----------------------*/
        /*Interrupt code here*/
        /*-----------------------*/

IEN0 |= 0x80;                  // Enable all interrupts
}
//---------------------------------------------------------------------------
```


1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Serial Interface

The VERSA MIX provides two asynchronous UART interfaces: UART 0 and UART 1. Each serial port has a 10-bit timer devoted to baud rate generation.

Serial Port Data Buffers

Both serial ports operate in full duplex asynchronous mode. The VERSA MIX buffers receive data in a holding register, enabling the UART to accept an incoming word before the software has read the previous value.

The buffers consist of two separate registers. The S1BUF register and the S0BUF register.

TABLE 59: (S0BUF) SERIAL PORT 0, DATA BUFFER - SFR 99H

7	6	5	4	3	2	1	0
S0BUF [7:0]							

TABLE 60: (S1BUF) SERIAL PORT 1, DATA BUFFER - SFR C1H

7	6	5	4	3	2	1	0
S1BUF [7:0]							

UART0 0: Operating Modes

The operation of the UART0 of the VERSA MIX is close to the standard 8051 UART. It can operate in four distinct modes as shown below:

TABLE 61: SERIAL PORT 0 MODES

SM0	SM1	MODE	DESCRIPTION	BAUD RATE
0	0	0	Shift Register	Fosc/12
0	1	1	8-bit UART	Variable
1	0	2	9-bit UART	Fclk/32 or /64
1	1	3	9-bit UART	Variable

**Note that the speed in mode 2 depends on SMOD bit in the Special Function Register PCON when SMOD = 1 fclk/32

The clock source for the UART0 can be derived from the Timer 1. However a dedicated 10bit baud rate generator is provided for UART0.

At the exception of the clock source and the actual baud rate configuration, the control of the UART0 is performed through the S0CON register described in detail below:

TABLE 62: (S0CON) SERIAL PORT 0, CONTROL REGISTER - SFR 98H

7	6	5	4
S0M0	S0M1	MPCE0	R0EN

3	2	1	0
T0B8	R0B8	T0I	R0I

Bit	Mnemonic	Function
7	S0M0	Sets Serial Port Operating Mode See Table 61 below
6	S0M1	
5	MPCE	1 = Enables the multiprocessor communication feature.
4	R0EN	1 = Enables serial reception. Cleared by software to disable reception.
3	T0B8	The 9th transmitted data bit in Modes 2 and 3. Set or cleared by the CPU, depending on the function it performs (parity check, multiprocessor communication etc.).
2	R0B8	In Modes 2 and 3, it is the 9th data bit received. In Mode 1, if sm20 is 0, RB80 is the stop bit. In Mode 0, this bit is not used. Must be cleared by software.
1	T0I	Transmit interrupt flag set by hardware after completion of a serial reception. Must be cleared by software.
0	R0I	Receive interrupt flag set by hardware after completion of a serial reception. Must be cleared by software.

UART0 - Mode 0

In this mode, pin RX0 is used as an input and an output, while TX0 is used only to output the shift clock. For an operation in this mode, 8 bits are transmitted with the LSB as the first bit. In addition, the baud rate is fixed at 1/12 of the crystal oscillator frequency. In order to initialize reception in this mode, the user must set the bits R0I and R0EN in the S0CON register to 0 and 1 respectively. Note that in other modes, when R0EN=1, the interface begins to receive data.

UART0 - Mode 1

In this mode, pin RX0 serves uniquely as an input and TX0 serves as a serial output. In this case, no external shift clock is used. In mode 0, 10 bit are transmitted:

- One logical low start bit;
- 8 bits of data starting with the LSB;
- One logical high stop bit.

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

During a reception, the start bit synchronizes the transmission and 8 bits of data are available by reading S0BUF. The reception is completed once the stop bit sets the R0B8 flag in the S0CON register.

UART0 - Mode 2

In this mode, pin RX0 is used as an input and as an output while TX0 is used only to output the shift clock. For an operation in this mode, 11 bits are transmitted or received. These 11 bits are composed of:

- One logical low start bit,
- 8 bits of data (LSB first),
- One programmable 9th bit,
- One logical high stop bit.

The purpose of the 9th bit is used to control the parity of the serial interface. For a data transmission, bit TB80 of the S0CON is outputted as the 9th bit. For a reception, the 9th bit will be stored into the RB80 bit of the S0CON register.

UART0 - Mode 3

Mode 3 is essentially identical to Mode 2. However, in Mode 3, the user may use the internal baud rate generator or timer 1 to set the baud rate.

UART0 - Baud Rate Generator

The UART0 baud rate clock can come from either Timer 1 or the dedicated 10bit baud rate generator

The selection between these two clocks source is made using the bit BAUDSRC of the U0BAUD register shown below:

TABLE 63: (U0BAUD) UART0 BAUD RATE SOURCE SELECT - SFR D8H

7	6	5	4	3	2	1	0
BAUDSRC	-	-	-	-	-	-	-

7	BAUDSRC	Baud rate generator clock source 0 = Timer 1 1 = Use UART0 dedicated Baud rate generator
6:0	-	-

The use of the UART0 dedicated baud rate generator permits to free up the Timer 1 for other applications. It is very simple to use as shown below:

The S0RELH and S0REL permit to store the 10 bit reload value of the UART0 baud rate generator.

TABLE 64: (S0RELL) SERIAL PORT 0, RELOAD REGISTER, LOW BYTE - SFR 96H

7	6	5	4	3	2	1	0
S0RELL [7:0]							

TABLE 65: (S0RELH) SERIAL PORT 0, RELOAD REGISTER, HIGH BYTE - SFR 97H

7	6	5	4	3	2	1	0
S0RELH [15:8]							

The reload value to put into the SOREL register can be calculated from the equations shown below:

Mode 3: For BAUDSRC=1

$$SOREL = 1024 - \frac{2^{SMOD} \times f_{clk}}{64 \times \text{Baud Rate}}$$

$$\text{Baud Rate} = \frac{2^{SMOD} \times f_{clk}}{64 \times (1024 - SOREL)}$$

TABLE 66: SERIAL 0 BAUD RATE SAMPLE VALUES BAUDSRC = 1, SMOD = 1

Desired Baud Rate	S0REL @ f_{clk}= 11.0592 MHz	S0REL @ f_{clk}= 14.746 MHz	S0REL @ f_{clk}= 16.000 MHz
115.2 kbps	3FDh	3FCh	N/A
57.6 kbps	3FAh	3F8h	N/A
19.2 kbps	3EEh	3E8h	3E6h
9.6 kbps	3DCh	3D0h	3CCh
2.4 kbps	370h	340h	330h
1.2 kbps	2E0h	280h	25Fh
300 bps	N/A	N/A	N/A

TABLE 67: SERIAL 0 BAUD RATE SAMPLE VALUES BAUDSRC =1, SMOD = 0

Desired Baud Rate	S0REL @ f_{clk}= 11.0592 MHz	S0REL @ f_{clk}= 14.746 MHz	S0REL @ f_{clk}= 16.000 MHz
115.2 kbps	N/A	3FEh	N/A
57.6 kbps	3FDh	3FCh	N/A
19.2 kbps	3F7h	3F4h	3F3h
9.6 kbps	3EEh	3E8h	3E6h
2.4 kbps	3B8h	3A0h	398h
1.2 kbps	370h	340h	330
300 bps	1C0h	100	0BFh

The Timer 1 can also be used as the baud rate generator for the UART0. Setting BAUDSRC to 0 assign Timer 1 output to UART0

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

When the baud rate clock source comes from Timer 1 the baud rate and the timer reload value can be calculated using the following formulas:

TABLE 68: EQUATION TO CALCULATE BAUD RATE FOR SERIAL 0

Serial 0: mode 1 and 3
Mode 1: ForU0BAUD.7=0 (standard mode)
$\text{Baud Rate} = \frac{2^{SMOD} \times f_{clk}}{32 \times 12 \times (256-TH1)}$
$TH1 = 256 - \frac{2^{SMOD} \times f_{clk}}{32 \times 12 \times \text{Baud Rate}}$

TABLE 69: UART 0 BAUD RATE SAMPLE VALUES BAUDSRC =0, SMOD = 1

Desired Baud Rate	TH1 @ f_{clk}= 11.0592 MHz	TH1 @ f_{clk}= 14.746 MHz	TH1 @ f_{clk}= 16.000 MHz
115.2 kbps	N/A	N/A	N/A
57.6 kbps	FFh	N/A	N/A
19.2 kbps	FDh	FCh	N/A
9.6 kbps	FAh	F8h	N/A
2.4 kbps	E8h	E0h	DDh
1.2 kbps	D0h	C0h	BBh
300 bps	40h	N/A	N/A

TABLE 70:UART 0 BAUD RATE SAMPLE VALUES BAUDSRC =0, SMOD = 0

Desired Baud Rate	TH1 @ f_{clk}= 11.0592 MHz	TH1 @ f_{clk}= 14.746 MHz	TH1 @ f_{clk}= 16.000 MHz
115.2 kbps	N/A	N/A	N/A
57.6 kbps	N/A	N/A	N/A
19.2 kbps	N/A	Feh	N/A
9.6 kbps	FDh	FCh	N/A
2.4 kbps	F4h	F0h	N/A
1.2 kbps	E8h	E0h	DDh
300 bps	A0h	80h	75h

Goal Semiconductor inc. provide on its web site software tools intended to permit calculation of the timers reload values for the VERSA MIX Serial ports.

Setting Up and Using UART 0

In order to use the UART0, the following operations must be performed:

- Enable the UART0 Interface
- Set I/O Pad direction TX= output, RX=Input
- Enable Reception
- Configure the Uart0 controller S0CON

Below is the Sample C code for the UART 0 interrupt setup and module configuration:

```
//-------------------------------------------------------------------------
// Sample C code to setup
//-------------------------------------------------------------------------

// UART 0 setup

IEN0 |= 0x80;                  // Enable all interrupts
IEN0 |= 0x10;                  // Enable interrupt UART 0
DIGPWREN |= 0x01;              // Enable UART 0
P3PINCFG = 0x01;               // Set pads direction
S0CON = 0x90;                  // Enable reception mode 2

// UART 0 example use

S0BUF = 0x99;                  // Send something on UART0
//-------------------------------------------------------------------------

// Interrupt function

void int_Uart0_0 (void)
{
IEN0 &= 0x7F;                  // Disable all interrupts

        /*-------------------------*/
        /* Interrupt code here*/
        /*-------------------------*/

if (S0CON&0x01==0x01)
        {
        S0CON &= 0xFE;    // Clear RI (it comes
                          // before T0I)
        }
        else
                {
                S0CON &= 0xFD;    //ClearT0I
                IEN0 |= 0x80;     //Enable all
                interrupts
                }
        }

//-------------------------------------------------------------------------
```


1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

UART 1: Operating Modes

The VERSA MIX UART1 provides two operating modes. These modes, named Mode A and Mode B provide the ability to perform transactions in 11bit and 10bit format respectively.

The UART1 Baud Rate generator comes exclusively from a dedicated 10bit baud rate generator.

The control of the UART1 is made using the S1CON Table 71 explains the bit functions of the Serial Port 1 control register.

TABLE 71: (S1CON) SERIAL PORT 1, CONTROL REGISTER - SFR C0H

7	6	5	4
S1M	-	MPCE1	R1EN

3	2	1	0
T1B8	R1B8	T1I	R1I

Bit	Mnemonic	Function
7	S1M	Operation mode Select
6	-	-
5	MPCE1	1 = Enables multiprocessor communication feature.
4	R1EN	If set, enables serial reception. Cleared by software to disable reception.
3	T1B8	The 9th transmitted data bit in mode A. Set or cleared by the CPU, depending on the function it performs (parity check, multiprocessor communication, etc.)
2	R1B8	In Mode A, it is the 9th data bit received. In Mode B, if SM21 is 0, RB81 is the stop bit. Must be cleared by software.
1	T1I	Transmit interrupt flag, set by hardware after completion of a serial transfer. Must be cleared by software
0	R1I	Receive interrupt flag, set by hardware after completion of a serial reception. Must be cleared by software

Below is a summary table of operating modes of the UART1.

TABLE 72: UART1 MODES

SM	MODE	DESCRIPTION	BAUD RATE
0	A	9-bit UART	Variable
1	B	8-bit UART	Variable

UART1 - Mode A

In this mode, 11 bits are transmitted or received. These 11 bits are composed of:

- One logical low start bit,
- 8 bits of data (LSB first),
- One programmable 9th bit,
- One logical high stop bit.

As in Mode 2 and 3 of UART0, this 9th bit is used for parity control. For a data transmission, bit TB81 of the S1CON is output as the 9th bit. For a reception, the 9th bit will be stored into the R1B8 bit of the S1CON register.

UART1 - Mode B

In this mode, 10 bits are transmitted and consist of:

- One logical low start bit;
- 8 bits of data starting with the LSB;
- One logical high stop bit.

During a reception, the start bit synchronizes the transmission and 8 bits of data are available by reading S1BUF. The reception is completed once the stop bit sets the R1B8 flag in the S1CON register.

UART1 - Baud Rate Generator

As mentioned earlier, the UART1 Baud Rate comes from a dedicated 10bit baud rate generator module.

The S1REL registers are used to adjust the baud rate on Serial 1.

TABLE 73: (S1RELL) UART1, RELOAD REGISTER, LOW BYTE - SFR BEH

7	6	5	4	3	2	1	0
S1RELL [7:0]							

TABLE 74: (S1RELH) UART 1, RELOAD REGISTER, HIGH BYTE - SFR BFH

7	6	5	4	3	2	1	0
S1RELH [7:0]							

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

The formulas to calculate both the baud Rate and the S1REL L and S1RELH value are given below:

Serial 1
$\text{Baud Rate} = \frac{f_{clk}}{32 \times (2^{10}\text{-S1REL})}$ Note: S1REL.9-0 = S1RELH.1-0 + S1RELL.7-0
$\text{S1REL} = 1024 - \frac{f_{clk}}{32 \times \text{Baud Rate}}$

TABLE 75: SERIAL 1 BAUD RATE SAMPLE VALUES

Desired Baud Rate	S1REL @ f_{clk}= 11.0592 MHz	S1REL @ f_{clk}= 14.746 MHz	S1REL @ f_{clk}= 16.000 MHz
115.2 kbps	3FDh	3FCh	N/A
57.6 kbps	3FAh	3F8h	N/A
19.2 kbps	3EEh	3E8h	3E6h
9.6 kbps	3DCh	3D0h	3CCh
2.4 kbps	370h	34Fh	330h
1.2 kbps	2E0h	280h	25Fh

Setting Up and Using UART 1

In order to use the UART0, the following operations must be performed:

- Enable the UART0 Interface
- Set I/O Pad direction TX= output, RX=Input
- Enable Reception
- Configure the Uart0 controller S0CON

Below is the Sample C code for the UART 0 interrupt setup and module configuration:

```
//-------------------------------------------------------------------------
// Sample C code to setup UART 1
//-------------------------------------------------------------------------

// UART 1 setup

IEN0 |= 0x80;                   // Enable all interrupts
IEN2 |= 0x01;                   // Enable interrupt UART 1
DIGPWREN = 0x02;                // Enable module UART 1
P0PINCFG = 0x04;                // Set pads direction
S1CON = 0x10;                   // Enable reception, Mode A


// UART 1 example use


S1RELL = 0xFE;                  // Set baud rate
S1RELH = 0x03;
S1BUF = 0x33;                   // Send something

//-------------------------------------------------------------------------

// Interrupt function

void int_serial_1 (void) interrupt 16
{
IEN0 &= 0x7F;                   // Disable all interrupts

        /*-----------------------*/
        /*Interrupt code here*/
        /*-----------------------*/

if (S1CON&0x01==0x01)
        {
        S1CON &= 0xFE;          // Clear RI (it comes
                                // before T1I)
        }
        else
        {
        S1CON &= 0xFD;          // Clear T1I
        }
        IEN0 |= 0x80;}          // Enable all interrupts
        }
}
/-------------------------------------------------------------------------
```

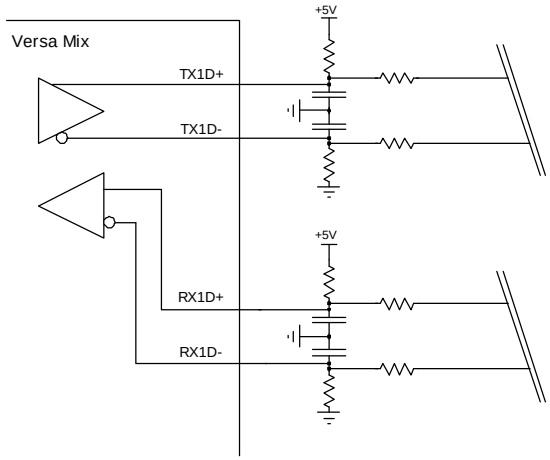

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

UART1 Differential Transceiver

The VERSA MIX includes a differential transceiver compatible with the J1708/RS-485/RS-422 standards. This Transceiver is intended to permit data transmission over long distance on twisted pair cable. The signals on the bus are differential permitting providing a high immunity to Electrical noise.

The common mode voltage range of the VERSA MIX's differential interface is -2.5V to 7.5V

FIGURE 22: J1708/RS-485/RS-422 COMPATIBLE INTERFACE (J1708 CONFIG)



The differential transceiver allows a number of devices to be connected on the bus.

The Differential transceiver I/Os are connected to the UART1 peripheral of the VERSA MIX. So the communication parameters such as the Data length, communication speed etc are managed by the UART1 interface.

Using the Uart1 Differential Transceiver

In order to use the Differential Transceiver interface, one must perform the following operations:

- Enable both the UART1 and the Differential interface by setting the bit 1 and Bit 2 of the DIGPWREN.
- Configure the operating mode of UART1 using the S1CON register.

Use UART1 S1BUF register to transmit / receive data through the differential transceiver.

When the transceiver is connected is connected in Half-Duplex mode, a careful management of the Uart1 interrupts will need to be performed, as every byte transmitted will generate a local interrupt if enabled.

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

SPI Interface

The VERSA MIX SPI interface is a very powerful and highly configurable interface that provide high speed data exchange with external devices such as High speed A/D, D/A, EEPROM etc.

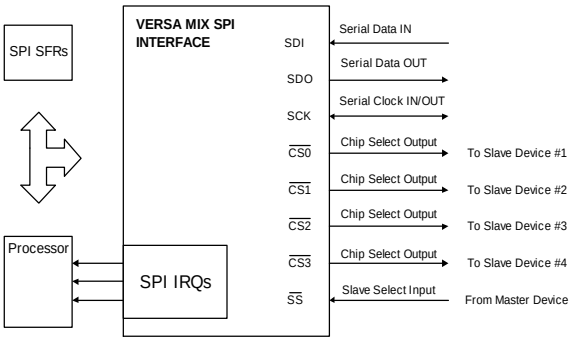
The SPI interface can operate as either a master or a slave device. In master mode, it can control up to 4 slave devices connected to the SPI bus.

The following list describes some of the features of the SPI interface:

- Permits synchronous serial data transfers
- Transaction size is configurable from 1-32 bits and more.
- Full duplex support
- SPI Modes 0, 1, 2, 3, and 4 supported (Full clock polarity and phase control)
- Up to four slave devices can be connected to the SPI bus when it is configured in master mode
- The SPI interface can be used in slave mode
- Data transmission speed is configurable
- Double 32bit buffers in transmission and reception
- 3 dedicated interrupt flags
  - TX-Empty
  - RX Data Available
  - RX Overrun
- Automatic/Manual control of the chip selects lines.
- SPI operation is not affected by the clock control unit

The following figure represents the SPI Interface in block diagram.

FIGURE 23: SPI INTERFACE BLOCK DIAGRAM



SPI transmit / receive buffer structure

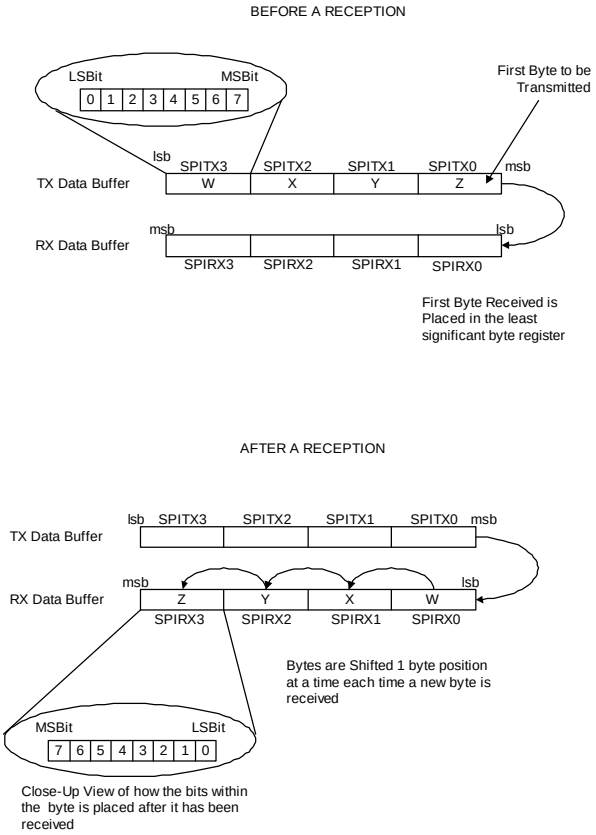
When receiving bytes, the first byte received is stored in the SPIRX0 Buffer. As bits continue to be received, bytes that were already in the buffer are shifted 1 byte position towards the least significant byte.

For example, lets suppose the SPI is about to receive 4 consecutive bytes of data: W,X,Y and Z. The first byte to be received is byte W. This byte will be placed in the register SPIRX0. At this point, the next byte to be received is byte X. The data buffer will shift the byte that is currently in the SPIRX0 register one byte position to the left, which is in this case SFR register SPIRX1. The new byte, X, will then be placed in SPIRX0. Following the same procedure, we see that bytes W,X,Y and Z will end up in RX data buffer registers SPIRX0.SPIRX1, SPIRX2 and SPIRX3 respectively.

The case where the SDO and SDI pins are shorted together is represented in the following diagram.

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

FIGURE 24: SPI INTERFACE RECEIVE TRANSMIT SCHEMATIC



When using the **SPI** Interface, it is important to keep in mind that a **transmission starts when the SPIRX3TX0 register is written into.**

From a SFR point of view, the transmission and the reception buffers of the SPI interface occupies the same addresses as shown below:

TABLE 76: (SPIRX3TX0) SPI DATA BUFFER, LOW BYTE - SFR E1H

7	6	5	4	3	2	1	0
SPIRX3TX0 [7:0]							

Bit	Mnemonic	Function
7-0	SPITX0	SPI Transmit Data Bits 7:0
	SPIRX3	SPI Receive Data Bits 31:24

TABLE 77: (SPIRX2TX1) SPI DATA BUFFER, BYTE 1 - SFR E2H

7	6	5	4	3	2	1	0
SPIRX2TX1 [15:8]							

Bit	Mnemonic	Function
15:8	SPITX1	SPI 1 Transmit Data Bits 15:8
	SPIRX2	SPI Receive 1 Data Bits 22:16

TABLE 78: (SPIRX1TX2) SPI DATA BUFFER, BYTE 2 - SFR E3H

7	6	5	4	3	2	1	0
SPIRX1TX2 [23:16]							

Bit	Mnemonic	Function
22:16	SPITX2	SPI Transmit Data Bits 22:16
	SPIRX1	SPI Receive Data Bits 15:8

Table 79: (SPIRX0TX3) SPI Data Buffer, High Byte - SFR E4h

7	6	5	4	3	2	1	0
SPIRX0TX3 [31:24]							

Bit	Mnemonic	Function
31:24	SPITX3	SPI Transmit Data Bits 31:24
	SPIRX0	SPI Receive Data Bits 7:0

SPI control registers

The following register summarizes the possible clock frequencies of the SPI interface.

TABLE 80: (SPICTRL) SPI CONTROL REGISTER - SFR E5H

7	6	5	4
SPICK [2:0]			SPICS_1

3	2	1	0
SPICS_0	SPICKPH	SPICKPOL	SPIMA_SL

Bit	Mnemonic	Function
7:5	SPICK	SPI Clock control 000 = OSC Ck Div 2 001 = OSC Ck Div 4 010 = OSC Ck Div 8 011 = OSC Ck Div 16 100 = OSC Ck Div 32 101 = OSC Ck Div 64 110 = OSC Ck Div 128 111 = OSC Ck Div 256
4:3	SPICS	Active CS line in Master Mode 00 = CS0- Active 01 = CS1- Active 10 = CS2- Active 11 = CS3- Active
2	SPICKPH	SPI Clock Phase
1	SPICKPOL	SPI Clock Polarity 0 – CK Polarity is Low 1 – CK Polarity is High
0	SPIMA_SL	Master / -Slave 1 = Master 0 = Slave

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

SPI Operating speed

Three bits of the SPICTRL register serves to adjust the communication speed of the SPI interface.

SPICK Div Ratio	Fosc = 16.00MHz	Fosc = 14.74MHz	Fosc = 11.059MHz
Clk Div 2	8 MHz	7.37 MHz	5.53 MHz
Clk Div 4	4 MHz	3.68 MHz	2.76 MHz
Clk Div 8	2 MHz	1.84 MHz	1.38 MHz
Clk Div 16	1 MHz	922 kHz	691 kHz
Clk Div 32	500 kHz	461 kHz	346 kHz
Clk Div 64	250 kHz	230 kHz	173 kHz
Clk Div 128	125 kHz	115 kHz	86 kHz
Clk Div 256	62 kHz	57.6 kHz	43.2 kHz

SPI Master Chip select control

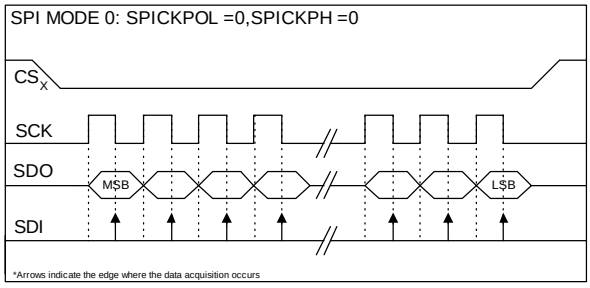
When the SPI is configured in Master Mode, The SPICS bits control which Slave device will be accessed by the SPI interface.

In the following sections we will show you how the SPI Clock Polarity and Phase affect the reading and writing operations of the SPI interface. Note that when the chip select bit (CSx) goes to 0, the SCK clock begins to operate. The difference in the operating modes lies in the SDO and SDI parameters.

SPI Mode 0

- Data is placed on the line (SDO) at every rising edge of the clock.
- Data is latched at every falling edge of the clock.

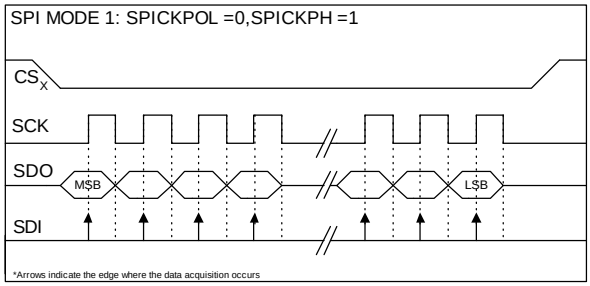
FIGURE 25: SPI MODE 0



SPI Mode 1

- Data is placed on the lines (SDO) at every falling edge of the clock.
- Data is latched at every rising edge of the clock.

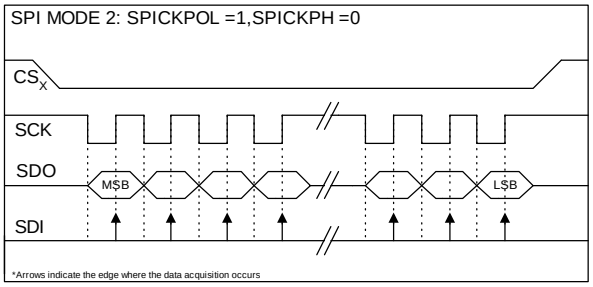
FIGURE 26: SPI MODE 1



SPI Mode 2

- Data is placed on the lines (SDO) at every falling edge of the clock.
- Data is latched at every rising edge of the clock.

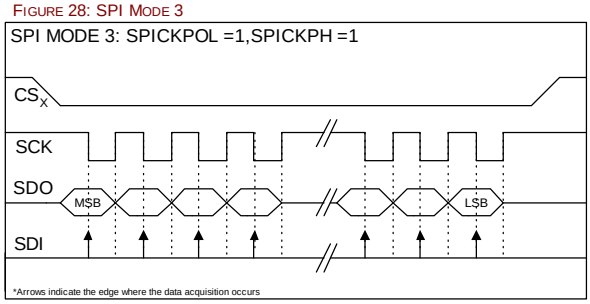
FIGURE 27: SPI MODE 2



SPI Mode 3

- Data is placed on the lines (SDO) at every rising edge of the clock.
- Data is latched at every falling edge of the clock.

FIGURE 28: SPI MODE 3



1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

SPI Interrupts

The SPI interface provides interrupt resources for three specific events that are:

- SPI Rx Overrun
- SPI RX Data Available
- SPI TX Empty

The SPIRXOVIE, SPIRXAVIE and SPITXEMPIE bits of the SPICONFIG register permits to individually enable those interrupts source at the SPI interface level.

At the processor level, two interrupts vectors are dedicated to the SPI interface:

- SPI RX data available and Overrun interrupt
- SPI Tx empty interrupt

In order to have the processor to jump to the associated interrupt routine, you must also enable one or both these interrupt in the IEN1 register as well as set the EA bit of the IEN0 register (see interrupt section).

TABLE 81: (SPICONFIG) SPI CONFIG REGISTER - SFR E6H

7	6	5	4
SPICSLO	-	FSONICS3	SPIOPT

3	2	1	0
-	SPIRXOVIE	SPIRXAVIE	SPITXEMP

Bit	Mnemonic	Function
7	SPICSLO	Manual CS up (Master mode) 0 = The CSx goes low when transmission begins and returns to high when it ends. 1 = The CSx stays low after transmission ends. The user must clear this bit for the CSx line to return high.
6	-	-
5	FSONCS3	This bit sends the frame select pulse on CS3.
4	SPILOAD	This bit sends load pulse on CS3.
3	-	-
2	SPIRXOVIE	SPI Receiver overrun interrupt enable.
1	SPIRXAVIE	SPI Receiver available interrupt enable.
0	SPITXEMPIE	SPI Transmitter empty interrupt enable.

The SPIIRQSTAT register contains the interrupts flags of the SPI interface. This make possible to define the exact source of a given SPI interrupt when an SPI RX OV is triggered.

Moreover, monitoring the different bits of this register permit to control the SPI interface in pooling mode when the interrupts are disabled.

TABLE 82: (SPIIRQSTAT) SPI INTERRUPT STATUS REGISTER - SFR E9H

7	6	5	4
-	-	SPITXEMPTO	SPISLAVESEL

3	2	1	0
SPISEL	SPIOV	SPIRAXV	SPITXEMP

Bit	Mnemonic	Function
7:6	Reserved-	-
5	SPITXEMPTO	Flag that indicates that we have not reloaded the transmit buffer fast enough (only used for packets greater than 32 bits.).
4	SPISLAVESEL	Slave Select " NOT" (SSN)
3	SPISEL	This bit is the result of the logical AND operation between CS0,CS1,CS2 and CS3. (Indicates if one chip is selected.)
2	SPIOV	SPI Receiver overrun
1	SPIRXAV	SPI Receiver available
0	SPITXEMP	SPI Transmit buffer is ready to receive mode data. It does not flag that the transmission is completed.

SPI Manual Chip select control

The SPICSLO bit of the SPICONFIG serves to prevent the active chip select line to return to an high level when the SPI transaction is completed in Master mode.

SPI Manual Load control

The SPI can generate a LOAD pulse on the CS3 pin when the SPILOAD bit is set. Some D/A converters require such a signal to latch the data that have been loaded into them.

This feature permits to easily handle that without having to use a separate I/O pin for that purpose.

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

SPI Frame Select control

It is also possible to generate a positive pulse on the CS3 pin of the SPI interface by setting the FSONCS3 bit of the SPICONFIG register.

Some DSP compatible devices require such a signal and this feature permit to easily handle that without having to use a separate I/O pin for that purpose.

When both SPILOAD and FSONCS3 are selected, the frame select pulse has priority.

SPI Transaction Size

The SPI Size control register permit to define the size of SPI transactions.

The SPI transaction size follows the following formulas:

For SPISIZE from 0 to 31:

SPI Transaction Size = [SPISIZE + 1]

For SPISIZE from 32 to 255:

SPI Transaction Size = [SPISIZE*8 - 216]

TABLE 83: (SPISIZECTRL) SPI SIZE CONTROL REGISTER - SFR E7H

7	6	5	4	3	2	1	0
SPISIZE							

Bit	Mnemonic	Function
7:0	SPISIZE	Value of the SPI packet size

Setting Up the SPI Interface

In order use the SPI Interface, one must first set up and configure the module. The following lines of code can be used to perform these tasks. The first part of the code is the interrupt setup and module configuration whereas the second part is the interrupt function.

```
//-------------------------------------------------------------------------
// SAMPLE C CODE FOR SPI RX SETUP
//-------------------------------------------------------------------------
// SPI_RX SETUP

IEN0 |= 0x80;                    // ENABLE ALL INTERRUPTS
IEN1 |= 0x04;                    // ENABLE INTERRUPT SPI
DIGPWREN = 0x08;                 // ENABLE SPI
P2PINCFG = 0x4F;                 // SET PADS DIRECTION
SPICONFIG = 0x06;                // ENABLE RX_AVAILABLE
                                 // RX_overrun
                                 // SPI_RX example use
SPIRX3TX0 = 0x33;                          // Send something
//-------------------------------------------------------------------------
// Interrupt Function
void int_3_spi_rx (void) interrupt 10
{
IEN0 &= 0x7F;                    // Disable all interrupts
            /*-------------------------*/
/* INTERRUPT CODE HERE*/
/*-------------------------*/
/*-------------------------------------------------------------------------*/
/* READ SPIIRQSTAT TO DETERMINE IF RX_AVAILABLE RX_OVERRUN
/*
/* READ SPIRXTX0 TO CLEAR RX_AVAILABLE AND RX_OVERRUN
/*-------------------------------------------------------------------------*/

IRCON &= 0xFB;                   // CLEAR FLAG (IEX3)
IEN0 |= 0x80;                    // ENABLE ALL INTERRUPTS
}

//-------------------------------------------------------------------
// Sample C code for SPI TX setup
//-------------------------------------------------------------------------
// SPI TX setup

IEN0 |= 0x80;                    // Enable all interrupts
IEN1 |= 0x02;                    // Enable SPTXTEMPTY interrupt
DIGPWREN = 0x08;                 // Enable SPI
P2PINCFG = 0x4F;                 // Set pads direction
SPICONFIG = 0x01;                // Enable TX_empty
//-------------------------------------------------------------------------
// Interrupt function

void int_2_spi_tx (void) interrupt 9
{
IEN0 &= 0x7F;                    // Disable all interrupts

            /*-------------------------*/
            /* Interrupt code here*/
            /*-------------------------*/

IRCON &= 0xFD;                   // Clear flag (IEX2)
IEN0 |= 0x80;                    // Enable all interrupts
}

//-------------------------------------------------------------------------
```


1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

I²C Interface

The VERSA MIX includes an I²C compatible interface that can be configured both in Master and Slave modes.

When PM pin = 0:

- Master/Slave configurable I²C interface
- Can be used to communicate with I²C devices or systems
- Many devices can share the I²C bus
- Adjustable communication speed (Up to 1MHz)
- Manual/Automatic acknowledge
- Tx Empty / Rx Data Available / Rx Overrun Flags

When PM pin = 1:

- The I²C interface serves to program the VERSA MIX Flash memory
- The I²C interface can serve to control the VERSA MIX peripherals
- The VERSA MIX processor is halted

The formula for calculating the I^2C clock frequency is the following:

$$I^2C\ Clk = \frac{f_{osc}}{[8 \times (I2CCLKCTRL)]}$$

(Note that the I^2C clock is outputted on the SCL line)

The following table gives some examples of I2C clock values that will be outputted on the SCL pin when using a 16MHz crystal oscillator.

I2CCLKCTRL Value	I2C Clock (SCL Value)
01h	1MHz
03h	500kHz
07h	250kHz
13h	100kHz
27h	50kHz
C7h	10kHz

TABLE 84: (I2CCLKCTRL) I2C CLOCK CONTROL - SFR DBH

7	6	5	4	3	2	1	0
I2CCLKCTRL [7:0]							

Bit	Mnemonic	Function
7:0	I2CCLKCTRL	I2C Clock speed control

TABLE 85: (I2CCONFIG) I2C CONFIGURATION - SFR DAH

7	6	5	4
I2CMASKID	I2CRXOVIE	I2CRXDAVIE	I2CTXEMPIE

3	2	1	0
I2CMANACK	I2CACKMODE	I2CMSTOP	I2CMASTER

Bit	Mnemonic	Function
7	I2CMASKID	This is used to mask the chip ID when you have only two devices. Therefore in a transaction, rather that receiving the chip ID first, you will receive the first packet of data.
6	I2CRXOVIE	I2C Receiver overrun interrupt enable
5	I2CRXDAVIE	I2C Receiver available interrupt enable
4	I2CTXEMPIE	I2C Transmitter empty interrupt enable
3	I2CMANACK	1= Manual acknowledge line goes to 0 0= Manual acknowledge line goes to 1
2	I2CACKMODE	Used only with Master Rx, Master Tx, and Slave Rx. 1= Manual Acknowledge on 0= Manual Acknowledge off
1	I2CMSTOP	I2C Master receiver stops at next acknowledge phase. (read during data phase)
0	I2CMASTER	I2C Master mode enable

TABLE 86: (I2CCHIPID) I2C CHIP ID - SFR DCH

7	6	5	4	3	2	1	0
I2CID [6:0]							I2CWID

Bit	Mnemonic	Function
7:1	I2CID	The value of this chip's ID
0	I2WID	(Used only in slave mode) Indicates that the last bit that this chip received should not have been received by this chip. This bit is "read only" and will only go low the next time this chip receives a bit that should be received by this chip.

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

TABLE 87: (I2CIRQSTAT) I2C INTERRUPT STATUS - SFR DDH

7	6	5	4
I2CGOTSTOP	I2CNOACK	I2CSDA	I2CDATACK

3	2	1	0
I2CIDLE	I2CRXOV	I2CRXAV	I2CTXEMP

Bit	Mnemonic	Function
7	I2CSGOTSTOP	This means that the slave has received a stop (this bit is read only). Reset only when the master begins a new transmission.
6	I2CNOACK	Flag that indicates that no acknowledge has been received. Is reset at the start of the next transaction
5	I2CSDA	Value of SDA line.
4	I2CDATACK	Data acknowledge phase.
3	I2CIDLE	Indicates that I2C is idle
2	I2CRXOV	I2C Receiver overrun
1	I2CRXAV	I2C Receiver available
0	I2CTXEMP	I2C Transmitter empty

TABLE88: (I2CRXTX) I2C DATA BUFFER - SFR DEH

7	6	5	4	3	2	1	0
I2CRXTX [7:0]							

Bit	Mnemonic	Function
7:0	I2CTX	I2C Data Receiver transmitter buffer

Setting Up the I²C Interface

In order to use the I²C, one must first set up and configure the module. The following lines of code can be used to perform these tasks. The first part of the code is the interrupt setup and module configuration whereas the second part is the interrupt function.

Sample C code to configure the I²C interface

```
//-------------------------------------------------------------------------
// Sample C code to setup for I2C setup
//-------------------------------------------------------------------------

// I2C setup

IEN0 |= 0x80;                    // Enable all interrupts
IEN1 |= 0x08;                    // Enable interrupt int_4
DIGPWREN = 0x10;                          // Enable I2C
I2CCONFIG = 0x71;                         // Enable interrupts,
                    // MASTER MODE
I2CCLKCTRL = 0x00;               // I2C CLOCK

//-------------------------------------------------------------------

// FUNCTION

VOID INT_4_I2C (VOID) INTERRUPT 11
{
  IEN0 &= 0x7F; // DISABLE ALL INTERRUPTS

            /*----------------------*/
            /* interrupt code here*/
            /*----------------------*/

/*-------------------------------------------------------------------------/*
Read SPIIRQSTAT to determine if TX_empty, RX_available /* or
RX_overrun.
/* Read I2CRXTX to clear RX_available and RX_overrun
/*-------------------------------------------------------------------------

IRCON &= 0xF7;                   // Clear flag (IEX4)
IEN0 |= 0x80;}                   // Enable all interrupts

//-------------------------------------------------------------------------
```


1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Analog Signal Path

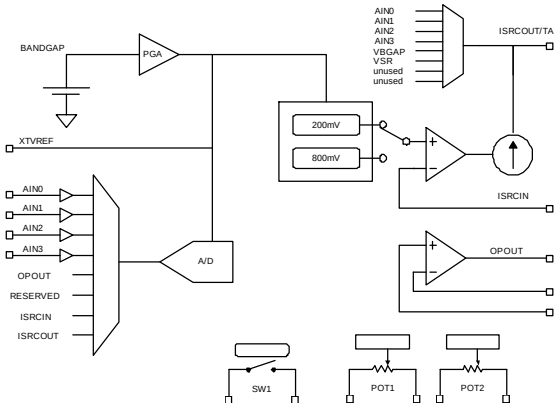
On the analog side, the VERSA MIX provides the following features:

- o 4 External input channels for A/D converter
- o Internal Band-Gap reference and PGA
- o Programmable current source
- o Input multiplexer
- o Multiplexed analog output
- o Digital potentiometers controlled by the µC
- o Digital switch controlled by the µC

This permits the VERSA MIX to be used as a single chip acquisition system.

Figure 29 shows the analog block of the VERSA MIX. The choice of an on-chip calibrated bandgap or an external reference provides the basis for all derived on-chip voltages. These voltages are used as the reference signals for the ADC and the current source.

FIGURE 29: ANALOG SIGNAL PATH OF THE VERSA MIX



The control of the internal and external reference, the multiplexer's current source drive, the ADC and their respective power downs are all done via the µ-Processor control through SFR registers.

TABLE 89: (INMUXCTRL) ANALOG INPUT MULTIPLEXER CONTROL REGISTER - SFR B5H

7	6	5	4	3	2	1	0
-	ADCINSEL [2:0]			AINEN [3:0]			

Bit	Mnemonic	Function
7	-	-
6:4	ADCINSEL	ADC Input Select 000 - AIN0 001 - AIN1 010 - AIN2 011 - AIN3 100 - OPOUT 101 - VSR 110 - ISRCIN 111 - ISRCOUT
3:0	AINEN[3:0]	Analog Input Enable

Table 90 summarizes the analog output multiplexer select line combinations and their corresponding outputs.

TABLE 90: (OUTMUXCTRL) ANALOG OUTPUT MULTIPLEXER CONTROL REGISTER - SFR B6H

7	6	5	4	3	2	1	0
-	-	-	-	-	TAOUTSEL [2:0]		

Bit	Mnemonic	Function
7:3	Unused	Unused
2:0	TAOUTSEL	Signal output on TA 000 – AIN0 001 – AIN1 010 – AIN2 011 – AIN3 100 – VBGAP 101 – VSR 110 – unused 111 – unused

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Internal Bandgap Reference and PGA

The VERSA MIX provides an internal bandgap reference coupled with a programmable gain amplifier. These two units provide a reference voltage for the ADC as well as the internal programmable current source. Both the bandgap and the PGA are calibrated during production and registers are automatically loaded when the device is reset.
However, different value can be written into the Bandgap and the PGA calibration registers at any time.

It is also possible to use an external reference instead of the internal bandgap.

TABLE 91: (BGAPCAL) BANDGAP CALIBRATION VECTOR REGISTER - SFR B3H

7	6	5	4	3	2	1	0
BGAPCAL [7:0]							

Bit	Mnemonic	Function
7:0	BGAPCAL	Bandgap data calibration

TABLE 92: (DPL1) PGA CALIBRATION VECTOR REGISTER - SFR B4H

7	6	5	4	3	2	1	0
BGAPCAL [7:0]							

Bit	Mnemonic	Function
7:0	PGACAL	8 MSBs of PGA Calibration Vector (LSB is on ISRCCAL1)

A/D Converter

The on-chip A/D converter can be configured for a number of different operating modes to meet the specific application requirements. It supports both continuous and single shot modes.

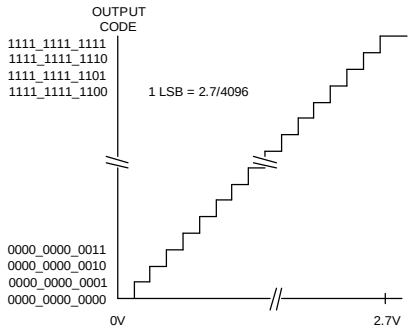
The continuous conversion mode sets the ADC to perform timed conversions of 1 or 4 channels, while the Single Shot conversion sets the ADC to perform one conversion of 1 or 4 channels at any given time.

When the ADC is configured to perform 4 channel conversions, the converted data will be available in a set of four 12-bit registers mapped into the processor's SFR space (ADC channels 0-3 registers).

When the ADC is configured to perform the conversion on one channel only, the conversion result is always placed into the ADC channel 0 register. This is why you will only be able to see one updated register when the ADC is configured to perform the conversion on only one channel. It is important to note that the four ADC Data registers are only used when the ADC is configured to perform sequential in four channel conversions.

The analog input voltage range for the ADC is 0 to 2.7V and the output coding is binary with 1 LSB = Full Scale/4096. The following figure describes the ideal transfer function for the ADC.

FIGURE 30: IDEAL A/D CONVERTER TRANSFER FUNCTION



1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

TABLE 93: (ADCCTRL) ADC CONTROL REGISTER - SFR A2H

7	6	5	4
ADCIRQCLR	XVREFCAP	CALSKIP	ADCIRQ

3	2	1	0
ADCIE	ONECHAN	CONT	ONESHOT

Bit	Mnemonic	Function
7	ADCIRQCLR	ADC interrupt clear Writing 1 Clears interrupt
6	XVREFCAP	Always keep this bit at 1
5	CALSKIP	1 = Skip Calibration Process 0 = Perform ADC Calibration
4	ADCIRQ	Read ADC Interrupt Flag Write 1 generate ADC IRQ
3	ADCIE	ADC interrupt enable
2	ONECHAN	1 = Conversion is performed on one channel Specified ADCINSEL 0 = Conversion is performed on 4 ADC channels
1	CONT	1 = Enable ADC continuous conversion
0	ONESHOT	1 = Force a single conversion on 1 or 4 channels

TABLE 94: (ADCCONVRLOW) ADC CONVERSION RATE REGISTER LOW BYTE - SFR A3H

Bit	Mnemonic	Function
7:0	CONVRLOW	Conversion rate low byte

TABLE 95: (ADCCONVERMED) ADC CONVERSION RATE REGISTER MED BYTE - SFR A4H

Bit	Mnemonic	Function
7:0	CONVRMED	Conversion rate medium byte

TABLE 96: (ADCCONVRHIGH) ADC CONVERSION RATE REGISTER HIGH BYTE - SFR A5H

Bit	Mnemonic	Function
7:0	CONVRHIGH	Conversion rate high byte

TABLE 97: (ADCD0LO) ADC CHANNEL 0 DATA REGISTER, LOW BYTE - SFR A6H

Bit	Mnemonic	Function
7:0	ADCD0LO	ADC channel 0 low

TABLE 98: (ADCD0HI) ADC CHANNEL 0 DATA REGISTER, HIGH BYTE - SFR A7H

Bit	Mnemonic	Function
3:0	ADCD0HI	ADC channel 0 high

TABLE 99: (ADCD1LO) ADC CHANNEL 1 DATA REGISTER, LOW BYTE - SFR A9H

7	6	5	4	3	2	1	0
ADCD1LO [7:0]							

Bit	Mnemonic	Function
7:0	ADCD1LO	ADC channel 1 low

TABLE 100: (ADCD1HI) ADC CHANNEL 1 DATA REGISTER, HIGH BYTE - SFR AAH

7	6	5	4	3	2	1	0
-	-	-	-	ADCD1HI [3:0]			

Bit	Mnemonic	Function
3:0	ADCD1HI	ADC channel 1 high

TABLE 101: (ADC2LO) ADC CHANNEL 2 DATA REGISTER, LOW BYTE - SFR ABH

7	6	5	4	3	2	1	0
ADCD2LO [7:0]							

Bit	Mnemonic	Function
7:0	ADCD2LO	ADC channel 2 low

TABLE 102: (ADCD2HI) ADC CHANNEL 2 DATA REGISTER, HIGH BYTE - SFR ACH

7	6	5	4	3	2	1	0
-	-	-	-	ADCD2HI [3:0]			

Bit	Mnemonic	Function
7:4	-	-
3:0	ADCD2HI	ADC channel 2 high

TABLE 103: (ADCD3LO) ADC CHANNEL 3 DATA REGISTER, LOW BYTE - SFR ADH

7	6	5	4	3	2	1	0
ADCD3LO [7:0]							

Bit	Mnemonic	Function
7:0	ADCD3LO	ADC channel 3 low

TABLE 104: (ADCD3HI) ADC CHANNEL 3 DATA REGISTER, HIGH BYTE - SFR AEH

7	6	5	4	3	2	1	0
-	-	-	-	ADCD3HI [3:0]			

Bit	Mnemonic	Function
7:4	-	-
3:0	ADCD3HI	ADC channel 3 high

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

A/D SFR Registers

Below is a list of the specifications and related formulas for each A/D SFR Register.

• ADC Clock Divider

The ADC Clock divider serve to derive the A/D reference clock from the system clock
The equation to get the ADC reference clock is given below.

Equation:

$$\text{ADC Clk ref} = \frac{f_{OSC}}{4x\,(ADCCDIV + 1)}$$

The ADC conversion requires 111 cycles to performing the conversion on one channel. Using the above equation, it is possible to derive the ADCCLKDIV register value to perform the conversion at a given rate.

The following table gives recommended ADCCLKDIV register value according to conversion rate. The number given are conservative figures and derived from a 14.74MHz clock

ADCCLKDIV	Maximum Conv. Rate
0x02	10500 Hz
0x03	8000 Hz
0x05	5000 Hz
0x07	4000 Hz
0x08	3500 Hz
0x09	3200 Hz
0x0B	2500 Hz
0x0D, 0x0E, 0x0F	2200 Hz

If you perform the conversion on 4 channels, divide the max conv., Rate by 4.

TABLE 105: (ADCCLKDIV) ADC CLOCK DIVISION CONTROL REGISTER - SFR 95H

7	6	5	4	3	2	1	0
ADCCLKDIV [7:0]							

Bit	Mnemonic	Function
7:0	ADCCLKDIV	ADC clock divider (R/W)

• A/D Conversion Rate Registers
  • 24-bit word sets the A/D conversion rate in continuous mode

Equation:

$$\text{Conversion rate registers value (24-bit)} = \frac{f_{OSC}}{Conv\_Rate}$$

• A/D Control Register
  • Enable control of the A/D
  • External ref cap select
  • Single shot/continuous operation mode select
  • 4-Channel/1-Channel control
  • A/D interrupt enable control/interrupts clear
  • A/D completion/interrupt flag

• A/D Input Mux Control
  • Selects which input is sampled in the 1-Channel mode
  • Individual control of input buffers

Setting up the A/D Converter

In order use the A/D converter, one must first set up and configure the module. The following lines of code can be used to perform these tasks. The first part of the code is the interrupt setup and module configuration whereas the second part is the interrupt function.

Sample C code to setup the A/D converter:

```
//-------------------------------------------------------------------------

// Sample C code to setup the Analog to Digital Converter
//-------------------------------------------------------------------------

//ADC Setup


IEN0 |= 0x80;                  // Enable all interrupts
IEN1 |= 0x20;                  // Enable interrupt int_6
ANALOGPWREN = 0x04;            // Enable ADC
ADCCLKDIV= 0x00;
ADCCONVRLOW= 0x10;
ADCCONVRMED = 0x00;
ADCCONVRHIGH = 0x00;
ADCCTRL = 0x09;                // One shot,

// Enable ADC Interrupt

//-------------------------------------------------------------------------

//Interrupt Function

void int_6_adc (void) interrupt 13
{
IEN0 &= 0x7F;                  // Disable all interrupts

        /*-----------------------*/
        /*Interrupt code here*/
        /*-----------------------*/

IRCON &= 0xDF;                 // Clear flag (ircon.5)
IEN0 |= 0x80;                  // Enable all interrupts
}

//-------------------------------------------------------------------------
```


1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Warning:
When using the ADC, makes sure the output multiplexer controlled by the TAEN bit of the ANALOGPWREN register (92h) is powered down in all times. Otherwise, it is likely that the signal present on the ISRCOUT will be routed back to the selected ADC input, causing signal reading error.

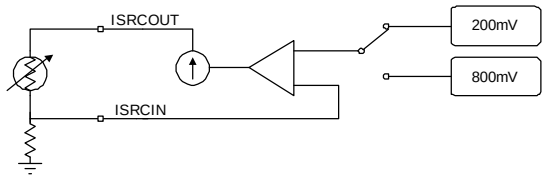
Programmable Current Source

The VERSA MIX includes a programmable current source intended to provide excitation to resistive sensors connected between the ISRCOUT and ISRCIN pins

To ensure current output stability, the current source provides a feedback input, ISRCIN.
The feedback is voltage controlled and can be dynamically set to either 200mV or 800mV.
Placing a resistor between the ISRC pin and the ground defines the output current of the current source.

It is possible to adjust the ISRC output to a current of up to 530µA.

FIGURE 31: PROGRAMMABLE CURRENT SOURCE TO EXCITE SENSOR



As shown in Figure 31, a resistive sensor must be connected between the ISRCOUT and the ISRCIN.

In order to perform A/D conversion of the voltage presents at the terminal of the current source. There is an internal link between each the ISRCOUT and ISRCIN pins and the Input multiplexer of the ADC converter.

TABLE 106: (ISRCCAL1) CURRENT SOURCE CALIBRATION VECTOR FOR 200MV FEEDBACK VALUE - SFR BCH

7	6	5	4	3	2	1	0
PGACAL0	ISRCCAL1 [6:0]						

Bit	Mnemonic	Function
7	PGACAL0	Bit 0 of PGACAL
6:0	ISRCCAL1	Calibration Value for ISRC feedback of 200mV

TABLE 107: (ISRCCAL2) CURRENT SOURCE CALIBRATION VECTOR FOR 800MV FEEDBACK VALUE - SFR BDH

7	6	5	4	3	2	1	0
-	ISRCCAL2 [6:0]						

Bit	Mnemonic	Function
7	-	-
6:0	ISRCCAL2	Calibration Value for ISRC feedback of 800mV

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Digital Potentiometers

The VERSA MIX also has two digital potentiometers that the user may control by setting the bits of either the DIGPOT1 register or the DIGPOT2 register.

Here is a short list of some possible applications for the digital potentiometers:

- Gain control
- Offset adjustment
- A/D input attenuation

TABLE 108: (DIGPOT1) DIGITAL POTENTIOMETER 1 CONTROL REGISTER - SFR BAH

7	6	5	4	3	2	1	0
DIGPOT1 [7:0]							

Bit	Mnemonic	Function
7-0	POT1RES	Potentiometer Value R_{pot} = (POT1RES / 256) x R_{max} (TBD)

TABLE 109: (DIGPOT2) DIGITAL POTENTIOMETER 2 CONTROL REGISTER - SFR BBH

7	6	5	4	3	2	1	0
DIGPOT2 [7:0]							

Bit	Mnemonic	Function
7-0	POT2RES	Potentiometer Value R_{pot} = (POT2RES / 256) x R_{max}

Operational Amplifier

The VERSA MIX is equipped with an operational amplifier. This op amp can be used for a wide array of analog applications. Here is a short list of some applications.

- Gain control
- Offset Control
- Reference buffering
- Integrator
- Other standard op amp applications

The op-amp on the VERSA MIX has an open-loop gain of 105 decibels, a unit gain bandwidth of 5MHz and it is able to drive a load of 5kO and 20pF. The slew rate of the Op-Amp is 5V/µs. The output voltage swing is between 0.1 and 4.9 Volts.

Digitally Controlled Switches

On the VERSA MIX, there are one digital switches, composed of 4 sub-switches connected in parallel. These sub-switches can be individually controlled by writing to the SFR register at B7h.

The switch-on resistance is between 50 and 100 Ohms depending on the number of sub-switches being used. If, for example, one sub-switch is closed, the switch resistance will be about 100 Ohms, and if all 4 switches are closed, the switch resistance will go down to about 50 Ohms.

TABLE 110: (SWITCHCTRL) USER SWITCHES CONTROL REGISTERS - SFR B7H

7	6	5	4	3	2	1	0
SWITCHCTRL [7:0]							

Bit	Mnemonic	Function
7:4		Not used
3:0	SW1CTRL	Switch 1 control (composed of 4 individual switches each bit controlled)

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Interrupts

The VERSA MIX is a highly integrated device incorporating a vast number of peripherals for which a comprehensive set of 12 interrupts + 1 reserved sources to ease system program development. Nearly all of the active peripherals in the VERSA MIX are able to generate a specific interrupt that can provide feedback to the MCU core that an event has occurred or a task has been completed.

Below is a list of things that one should know about the interrupts on the VERSA MIX:

- Each digital peripheral on the VERSA MIX has an interrupt channel.
- The SPI, UARTs and I²C all have event specific flag bits.
- When the processor is in IDLE mode, an interrupt may be used to wake it up.
- The processor can run at full speed during interrupt routines.

The following table summarizes the interrupt sources, the natural priority and the associated interrupt vector addresses.

TABLE 111: INTERRUPT SOURCES AND NATURAL PRIORITY

Interrupt	Interrupt Vector
Reserved	0E43h
INT0	0003h
UART1	0083h
TIMER 0	000Bh
SPI Tx	004Bh
INT1	0013h
SPI RX & SPI RX OVERRUN	0053h
TIMER 1	001Bh
I2C (Tx, Rx, Rx Overrun)	005Bh
UART0	0023h
MAC OVERFLOW BIT	0063h
TIMER 2	002Bh
A/D CONVERTER and Port	006Bh

The SPI, UARTs and I²C interfaces have event specific flag bits.

It is also possible to program the interrupts to wake-up the processor from IDLE mode or force it to return to full speed when an interrupt occurs.

Interrupt Enable Registers

The following tables are the interrupt enable register representations and their bit functions:

TABLE 112: (IEN0) INTERRUPT ENABLE REGISTER 0 - SFR A8H

7	6	5	4
EA	WDT	T2IE	S0IE

3	2	1	0
T1IE	INT1IE	T0IE	INT0IE

Bit	Mnemonic	Function
7	EA	General Interrupt control 0 = Disable all Enabled interrupts 1 = Authorize all Enabled interrupts
6	WDT	Watchdog timer refresh flag. This bit is used to initiate a refresh of the watchdog timer. In order to prevent an unintentional reset, the watchdog timer the user must set this bit directly before SWDT.
5	T2IE	Timer 2 Overflow / external Reload interrupt 0 = Disable 1 = Enable
4	S0IE	Uart0 interrupt. 0 = Disable 1 = Enable
3	T1IE	Timer 1 overflow interrupt 0 = Disable 1 = Enable
2	INT1IE	External Interrupt 1 0 = Disable 1 = Enable
1	T0IE	Timer 0 overflow interrupt 0 = Disable 1 = Enable
0	INT0IE	External Interrupt 0 0 = Disable 1 = Enable

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

TABLE 113: (IEN1) INTERRUPT ENABLE 1 REGISTER -SFR E8H

7	6	5	4
T2EXIE	SWDT	ADCPCIE	MACOVIE

3	2	1	0
I2CIE	SPIOVIE	SPITEIE	reserved

Bit	Mnemonic	Function
7	T2EXIE	T2EX interrupt 0 = Disable 1 = Enable
6	SWDT	Watchdog timer start/refresh flag. Set to activate/refresh the watchdog timer. When directly set after setting WDT, a watchdog timer refresh is performed. Bit SWDT is reset.
5	ADCPCIE	ADC and Port change interrupt 0 = Disable 1 = Enable
4	MACOVIE	MAC Overflow 32 bits interrupt 0 = Disable 1 = Enable
3	I2CIE	I2C Interrupt 0 = Disable 1 = Enable
2	SPIOVIE	SPI Overrun interupt 0 = Disable 1 = Enable
1	SPITEIE	SPI Tx Empty interrupt 0 = Disable 1 = Enable
0	reserved	

TABLE 114: (IEN2) INTERRUPT ENABLE 2 REGISTER - SFR 9AH

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	S1IE

Bit	Mnemonic	Function
7-1	-	-
0	S1IE	Uart 1 Interrupt 0 = Disable Uart 1 Interrupt 1 = Enable Uart 1 Interrupt

TABLE 115: (PORTIRQEN) PORT CHANGE IRQ CONFIGURATION - SFR 9FH

7	6	5	4
P07IEN	P06IEN	P05IEN	P04IEN

3	2	1	0
P03IEN	P02IEN	P01IEN	P00IEN

Bit	Mnemonic	Function
7	P07IEN	Port 0.7 IRQ on change enable 0 = Disable 1 = Enable
6	P06IEN	Port 0.6 IRQ on change enable 0 = Disable 1 = Enable
5	P05IEN	Port 0.5 IRQ on change enable 0 = Disable 1 = Enable
4	P04IEN	Port 0.4 IRQ on change enable 0 = Disable 1 = Enable
3	P03IEN	Port 0.3 IRQ on change enable 0 = Disable 1 = Enable
2	P02IEN	Port 0.2 IRQ on change enable 0 = Disable 1 = Enable
1	P01IEN	Port 0.1 IRQ on change enable 0 = Disable 1 = Enable
0	P00IEN	Port 0.0 IRQ on change enable 0 = Disable 1 = Enable

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Interrupt Status Flags

TABLE 116: (IRCON) INTERRUPT REQUEST CONTROL REGISTER - SFR 91H

7	6	5	4
EXF2IF	TF2IF	ADCIF	MACIF

3	2	1	0
I2CIF	SPIRXIF	SPITXIF	Reserved

Bit	Mnemonic	Function
7	EXF2IF	Timer 2 external reload flag This bit informs the user whether an interrupt has been generated from T2EX, if the EXEN2 is enabled.
6	TF2IF	Timer 2 overflow flag
5	ADCIF	A/D converter interrupt request flag
4	MACIF	MAC unit interrupt request flag
3	I2CIF	I^2C interrupt request flag
2	SPIRXIF	RX available flag
1	SPITXIF	TX empty flag
0	Reserved	reserved

The other interrupt registers are summarized in the tables below:

TABLE 117: (PORTIRQSTAT) PORT CHANGE IRQ STATUS - SFR A1H

7	6	5	4
P07ISTAT	P06ISTAT	P05ISTAT	P04ISTAT

3	2	1	0
P03ISTAT	P02ISTAT	P01ISTAT	P00ISTAT

Bit	Mnemonic	Function
7	P07ISTAT	Port 0.7 changed 0 = No 1 = Yes
6	P06ISTAT	Port 0.6 changed 0 = No 1 = Yes
5	P05ISTAT	Port 0.5 changed 0 = No 1 = Yes
4	P04ISTAT	Port 0.4 changed 0 = No 1 = Yes
3	P03ISTAT	Port 0.3 changed 0 = No 1 = Yes
2	P02ISTAT	Port 0.2 changed 0 = No 1 = Yes
1	P01ISTAT	Port 0.1 changed 0 = No 1 = Yes
0	P00ISTAT	Port 0.0 changed 0 = No 1 = Yes

Interrupt Priority Registers

TABLE 118: (IP0) INTERRUPT PRIORITY REGISTER 0 - SFR B8H

7	6	5	4	3	2	1	0
UF8	WDTS	IP0 [5:0]					

Bit	Mnemonic	Function		
7	UF8			
6	WDTS	Watchdog timer status flag. Set by hardware when the watchdog timer was started. Can be read by software.		
5	IP0.5	Timer 2 Interrupt	-	External Interrupt 6
4	IP0.4	Serial Channel 0 Interrupt	-	External Interrupt 5
3	IP0.3	Timer 1 Interrupt	-	External Interrupt 4
2	IP0.2	External Interrupt 1	-	External Interrupt 3
1	IP0.1	Timer 0 Interrupt	-	External Interrupt 2
0	IP0.0	External Interrupt 0	Serial Channel 1 Interrupt	A/D Converter Interrupt

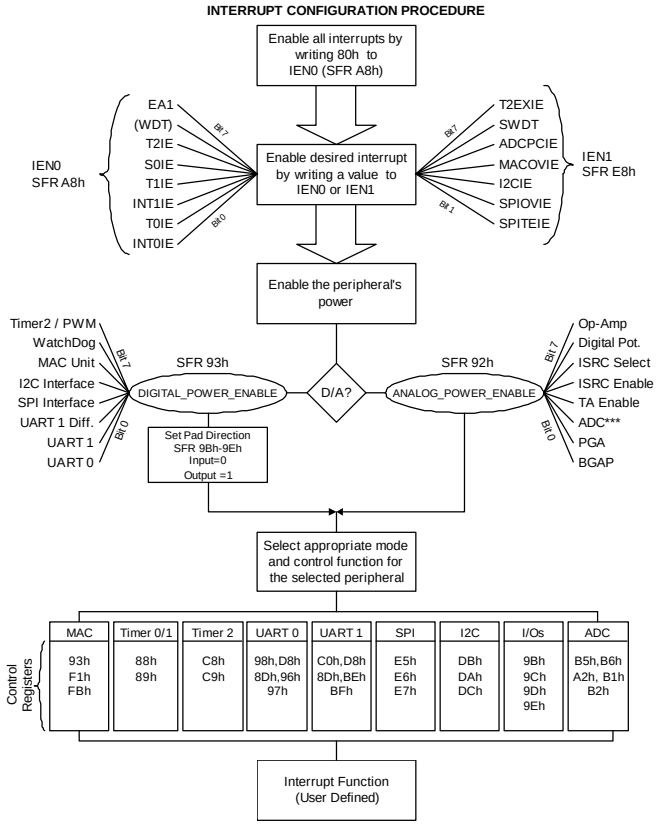
Table 119: (IP1) Interrupt Priority Register 1 - SFR B9h

7	6	5	4	3	2	1	0
-	-	IP1 [5:0]					

Bit	Mnemonic	Function		
7	-	-		
6	-	-		
5	IP1.5	Timer 2 Interrupt	-	External Interrupt 6
4	IP1.4	Serial Channel 0 Interrupt	-	External Interrupt 5
3	IP1.3	Timer 1 Interrupt	-	External Interrupt 4
2	IP1.2	External Interrupt 1	-	External Interrupt 3
1	IP1.1	Timer 0 Interrupt	-	External Interrupt 2
0	IP1.0	External Interrupt 0	Serial Channel 1 Interrupt	A/D Converter Interrupt

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

The following diagram provide a summary of the steps needed to configure the interrupts



Setting up INT0 and INT1 Interrupts

The IT0 and IT1 bit of the TCON register define if the external interrupt 0 and 1 will be edge or level triggered.
When an interrupt condition occurs on INT0 or INT1, the associated interrupt flag IE0 or IE1 is set. The interrupt flag is automatically cleared when the interrupt is serviced

TABLE 120: (TCON) TIMER 0, TIMER 1 TIMER/COUNTER CONTROL - SFR 88H

7	6	5	4
TF1	TR1	TF0	TR0

3	2	1	0
IE1	IT1	IE0	IT0

Bit	Mnemonic	Function
7	TF1	Timer 1 overflow flag set by hardware when Timer 1 overflows. This flag can be cleared by software and is automatically cleared when interrupt is processed.
6	TR1	Timer 1 Run control bit. If cleared Timer 1 stops.
5	TF0	Timer 0 overflows flag set by hardware when Timer 0 overflows. This flag can be cleared by software and is automatically cleared when interrupt is processed.
4	TR0	Timer 0 Run control bit. If cleared timer 0 stops.
3	IE1	Interrupt 1 edge flag. Set by hardware when falling edge on external INT1 is observed. Cleared when interrupt is processed.
2	IT1	Interrupt 1 type control bit. Selects falling edge or low level on input pin to cause interrupt.
1	IE0	Interrupt 0 edge flag. Set by hardware when falling edge on external pin INT1 is observed. Cleared when interrupt is processed.
0	IT0	Interrupt 0 type control bit. Selects falling edge or low level on input pin to cause interrupt.

In order use INT1 interrupts, one must first set up and configure the module.

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

INT0 example

Below is the Sample C code for the INT0 interrupt setup and module configuration:

```
//-----------------------------------------------------------------------
// Sample C code to setup INT0
//-----------------------------------------------------------------------


// INT0 setup

IEN0 |= 0x80;                    // Enable all interrupts
IEN0 |= 0x01;                    // Enable interrupt INT0


//-----------------------------------------------------------------------

// Interrupt Function

void int_ext_0 (void) interrupt 0
{
IEN0 &= 0x7F;                    // Disable all interrupts

        /*-----------------------*/
        /*Interrupt code here*/
        /*-----------------------*/

IEN0 |= 0x80;                    // Enable all interrupts
}

//-----------------------------------------------------------------------
```

INT1 example

The following code example show the INT1 interrupt setup and module configuration:

```
//-----------------------------------------------------------------------
// Sample C code to setup INT1
//-----------------------------------------------------------------------

// INT1 setup

IEN0 |= 0x80;                    // Enable all interrupts
IEN0 |= 0x04;                    // Enable interrupt INT1

//-----------------------------------------------------------------------

// Interrupt function

void int_ext_1 (void) interrupt 2
{
IEN0 &= 0x7F;                    // Disable all interrupts

        /*-----------------------*/
        /*Interrupt code here*/
        /*-----------------------*/

IEN0 |= 0x80;                    // Enable all interrupts
}
```


1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

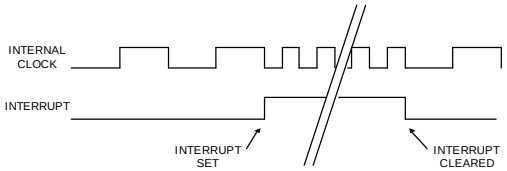
Clock Control Circuitry

The VERSA MIX clock control circuitry serves to dynamically adjust the clock speed of the VERSA MIX.

General Clock Control Unit

- Permits dynamic adjustment of system clock
- Serves to lower power consumption
- Possible clock rates: Fosc/1 down to Fosc/512
- Can make the clock return to full speed once an interrupt occurs

FIGURE 32: CLOCK TIMING WHEN AN INTERRUPT OCCURS



Processor Specific Power Saving Modes

IDLE Mode:

- µP clock is stopped
- The internal clock and enabled peripherals continue to run
- Any enabled interrupts can make the µP exit IDLE mode

Stop Mode:

- uP clock is stopped
- All internal clocking is stopped
- The following events can restart the processor: Reset, INT0, INT1, Debug Port, SPI Rx/Rx Overrun, and I²C

TABLE 121: (CLKDIVCTRL) CLOCK DIVISION CONTROL REGISTER -SFR 94H

7	6	5	4
SOFTRST	-	-	IRQNORMSPD

3	2	1	0
MCKDIV [3:0]			

Bit	Mnemonic	Function
7	SOFTRST	Writing 1 into this bit location provokes a reset. Read as a 0
6:5	-	-
4	IRQNORMSPD	0 = Full Speed in IRQ 1 = Selected speed during IRQs
3:0	MCKDIV	Master Clock Divisor 0000 – Sys CLK 0001 = SYS /2 0010 = SYS /4 0011 = SYS /8 0100 = SYS /16 0101 = SYS /32 0110 = SYS /64 0111 = SYS /128 1000 = SYS /256 1001 = SYS /512 (…) 1111 = SYS /32768

TABLE 122: (CKCON) CLOCK CONTROL - SFR 8EH

7	6	5	4	3	2	1	0
-	-	-	-	-	STRETCH [2:0]		

Bit	Mnemonic	Function
7:3	-	-
2:0	Stretch	Used for changing the stretch value for the read and write signal widths.

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Reset and Power Control

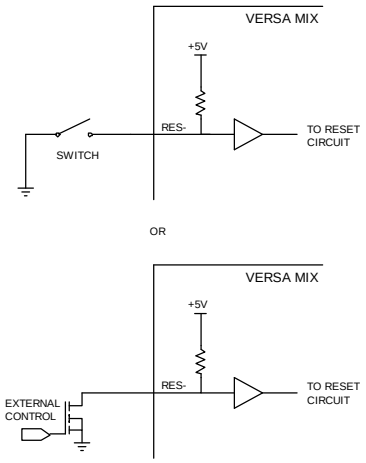
Reset Control

The VERSA MIX provides two resets, *POR_n* and RES-. *POR_n* is the power-on reset that is generated internally by the Power-On-Reset/Brown-Out device.

- Upon Reset, all SFR locations return to their default values and peripherals are disabled
- The internal 256 byte RAM cells are reset to 00h
- The external 1K RAM memory block is not affected by reset
- The VERSA MIX requires no external component for Reset (on power-up)
- No debouncing circuit is required when a manual Reset switch is used
- An internal pull-up is provided on the reset input

It is also possible to control the VERSA MIX reset line externally as shown on the following figures.

FIGURE 33: EXTERNAL RESET CONTROL



Power Control

The power management unit has two modes of operation: IDLE mode and STOP mode.

Idle Mode

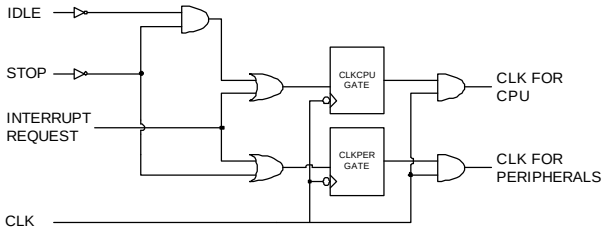
In order to enter IDLE mode, the user must set the IDLE bit of the PCON register (Table 125).

When the VERSA MIX is in IDLE mode, the internal clock and peripherals are still running. Power consumption drops because the CPU is not active. As soon as an interrupt or reset occurs, the CPU exits the IDLE state. To exit Idle mode, the 8051 does not require that the reset and interrupt signals 0/1 be level.

Stop Mode

In order to enter STOP mode, the user must set the STOP bit of the PCON register. In this mode, in contrast to IDLE mode, all internal clocking shuts down. The CPU will exit this state only when a no-clocked external interrupt or reset occurs (internal interrupts are not possible because they require clocking activity). In order to wake up from Stop mode, the 8051 requires that the reset and interrupt signals 0/1 be level.

FIGURE 34: POWER MANAGEMENT ON THE VERSA MIX



Represented below is the power control register of the VERSA MIX.

TABLE 123: (PCON) POWER CONTROL (CPU) - SFR 87H

7	6	5	4	3	2	1	0
SMOD_0	-	-	-	GF1	GF0	STOP	IDLE

Bit	Mnemonic	Function
7	SMOD	The speed in Mode 2 of Serial port 0 is controlled by this bit. When SMOD= 1, f_{clk} /32. This bit is also significant in Mode 1 and 3, as it adds a factor of 2 to the baud rate.
6	-	-
5	-	-
4	-	-
3	GF1	Not used for power management
2	GF0	Not used for power management
1	STOP	Stop mode control bit. Setting this bit turns on the STOP Mode. STOP bit is always read as 0.
0	IDLE	IDLE mode control bit. Setting this bit turns on the IDLE mode. IDLE bit is always read as 0.

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Watchdog Timer

The watchdog timer is a 15-bit counter. It consists of WDTH and WDTL: These two internal registers are used to keep track of the count of the watchdog timer. On the watchdog timer, the counter is incremented once every 24 or 384 clock cycles. Upon a reset, the watchdog is disabled and all registers are set to zero.

From the schematic in Figure 32, we notice that the watchdog timer consists of a 15-bit counter WDTL/H, a reload register WDTREL, prescalers that divide one twelfth of the oscillator by 2 or 16, and some control logic.

Start Procedure

In order for the watchdog to begin counting, the user must set bit 6 (WDOGEN) of DIGPWREN (Table 12).

To start the watchdog timer using the hardware automatic start procedure, the WDTS (IEN1) and WDT (IEN0) bit must be set. The watchdog will begin to run with default setting i.e. all registers will be set to zero.

When the WDT registers enter the state 7CFFh, the asynchronous signal, WDTS, will become active. This signal will set bit 6 in the IP0 register and trigger a reset.

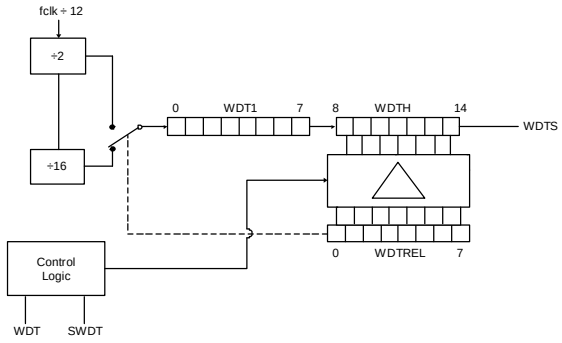
To prevent the WDT from resetting the VERSA MIX, you must clear it periodically by clearing the WDT bit of IEN0 register.

TABLE 124: (WDTREL) WATCHDOG TIMER RELOAD REGISTER - SFR D9H

7	6	5	4	3	2	1	0
PRES	WDTREL [6:0]						

Bit	Mnemonic	Function
7	PRES	Prescaler select bit. When set, the watchdog is clocked through an additional divide-by-16 prescaler.
6-0	WDTREL	7-bit reload value for the high-byte of the watchdog timer. This value is loaded to the WDT when a refresh is triggered by a consecutive setting of bits WDT and SWDT.

FIGURE 35: WATCHDOG TIMER

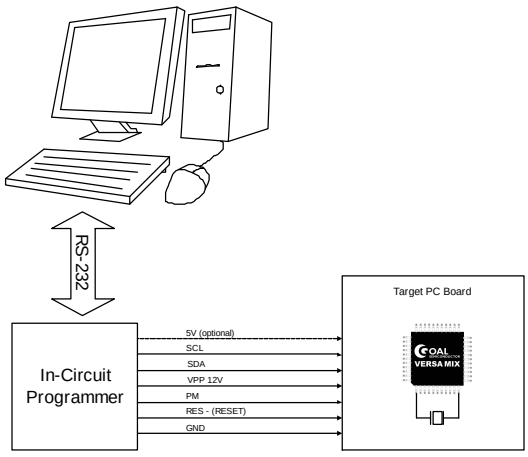


VERSA MIX Programming

When the PM pin is set to 1, the I²C interface serves to program the Flash memory of the VERSA MIX.

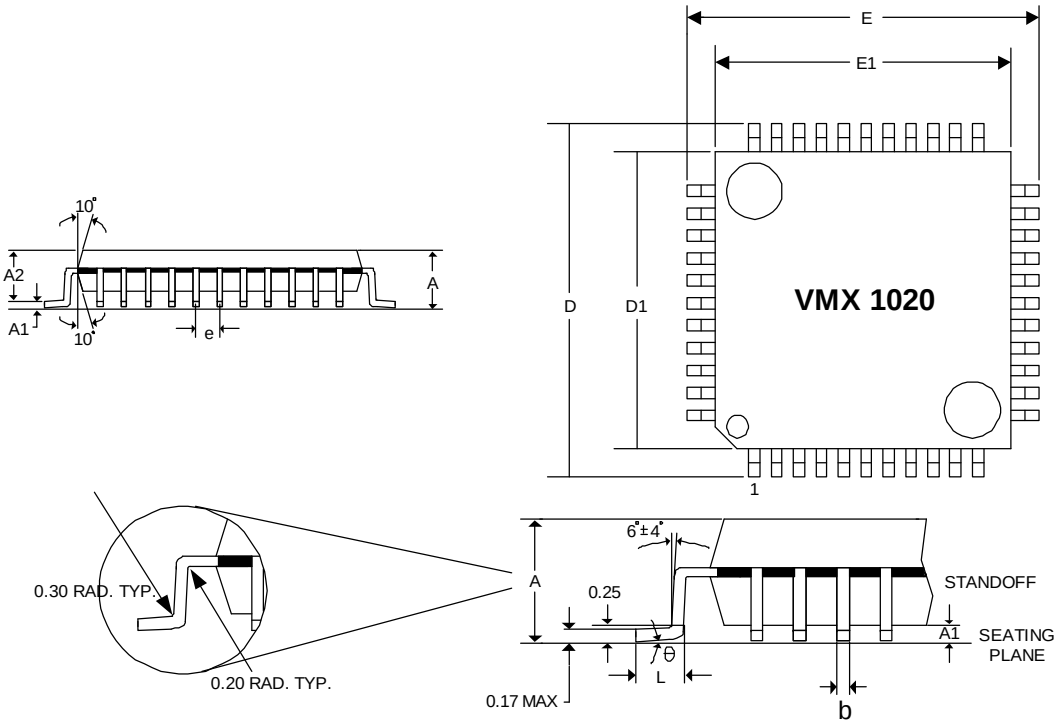
- The VERSA MIX Flash memory is programmed using the I²C interface in slave mode and control lines
- In-circuit programming interface is easy to implement at the board level
- An external programming voltage of 12V is required (provided by programmer)
- The VERSA MIX can be programmed using the DevBoard or the In-Circuit programmer

FIGURE 36: VERSA MIX PROGRAMMING



1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com

Plastic Quad Flat Package



		BODY +3.20mm Footprint
PACKAGE THICKNESS		2.00
Dims.	LEADS / TOLS.	64L
A	MAX.	2.35
A1		0.25 MAX
A2	+.10/-.05	2.00
D	±.25	17.20
D1	±.10	14.00
E	±.25	17.20
E1	±.10	14.00
L	+.15/-.10	.88
e	BASIC	.80
b	±.05	.35
θ		0º-7º

NOTES:

1) ALL DIMENSIONS ARE IN MILLIMETERS.

2) DIMENSIONS SHOWN ARE NOMINAL WITH TOLERANCES AS INDICATED.

3) FOOT LENGTH "L" IS MEASURED AT GAGE PLANE, 0.25 ABOVE SEATING PLANE.

Disclaimers

Right to make change - Goal Semiconductor reserves the right to make changes to its products - including circuitry, software and services - without notice at any time. Customers should obtain the most current and relevant information before placing orders.
Use in applications - Goal Semiconductor assumes no responsibility or liability for the use of any of its products, and conveys no license or title under any patent, copyright or mask work right to these products and makes no representations or warranties that these products are free from patent, copyright or mask work right infringement unless otherwise specified. Customers are responsible for product design and applications using Goal Semiconductor parts. Goal Semiconductor assumes no liability for applications assistance or customer product design.
Life support – Goal Semiconductor products are not designed for use in life support systems or devices. Goal Semiconductor customers using or selling Goal products for use in such applications do so at their own risk and agree to fully indemnify Goal Semiconductor for any damages resulting from such applications.

1134 Ste-Catherine St. W, Suite 900, Montreal, Quebec, Canada H3B 1H4 ✍ Tel: (514) 871-2447 ✍ http://www.goalsemi.com