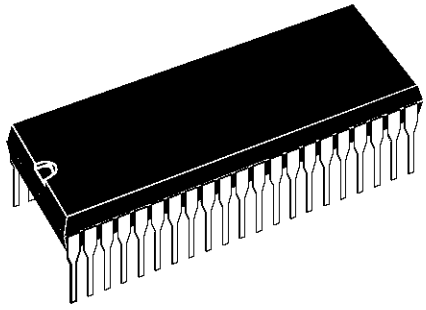
SGS-THOMSON MICROELECTRONICS

ST9291

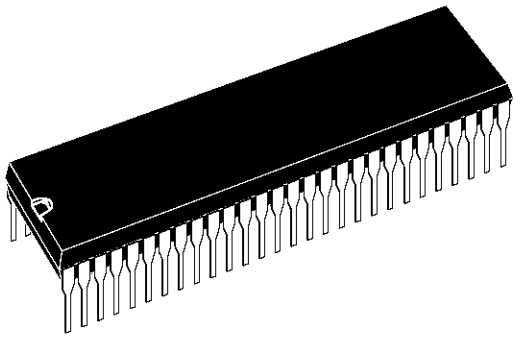
16-48K ROM HCMOS MCU WITH ON SCREEN DISPLAY AND VOLTAGE TUNING OUTPUT

FUNCTIONAL DESCRIPTION

- Register oriented 8/16 bit CORE with RUN, WFI and HALT modes
- Minimum instruction cycle time: 500ns (12MHz internal)
- 16 to 48K bytes of ROM, 384/640 bytes of RAM, 224 general purpose registers available as RAM, accumulators or index registers (Register File)
- 42-lead Shrink DIP package or 56-lead Shrink DIP package
- Interrupt handler and Serial Peripheral Interface as standard features
- 31 (42 pin package) / 42 (56 pin package) fully programmable I/O pins
- 34 character x15 rows software programmable On Screen Display module with colour, italic, underline, flash, transparent and fringe attribute options
- 14-bit Voltage Synthesis for tuning reference voltage.
- 8 8-bit PWM D/A outputs with repetition frequency 2 to 32kHz and 12V Open Drain Capability
- 16 bit Timer with 8 bit Prescaler, able to be used as a Watchdog Timer
- 16-bit programmable Slice Timer with 8-bit prescaler
- 3 channel Analog to Digital Converter, with integral sample and hold, fast 5.75µs conversion time, 6-bit guaranteed resolution
- Rich Instruction Set and 14 Addressing modes
- Division-by-Zero trap generation
- Versatile Development tools, including assembler, linker, C-compiler, archiver, graphic oriented debugger and hardware emulators
- Real Time Operating System
- Windowed EPROM parts available for prototyping and pre-production development phases



PSDIP42



PSDIP56

(Ordering Information at the end of the Datasheet)

DEVICE SUMMARY

Device	ROM	RAM	PACKAGE
ST9291J2/N2	16K	384	PSDIP42/56
ST9291J3/N3	16K	640	PSDIP42/56
ST9291J4/N4	24K	384	PSDIP42/56
ST9291J5/N5	24K	640	PSDIP42/56
ST9291J6/N6	32K	640	PSDIP42/56
ST9291J7/N7	48K	640	PSDIP42/56

Figure 1. 42 Pin Shrink DIP Pinout

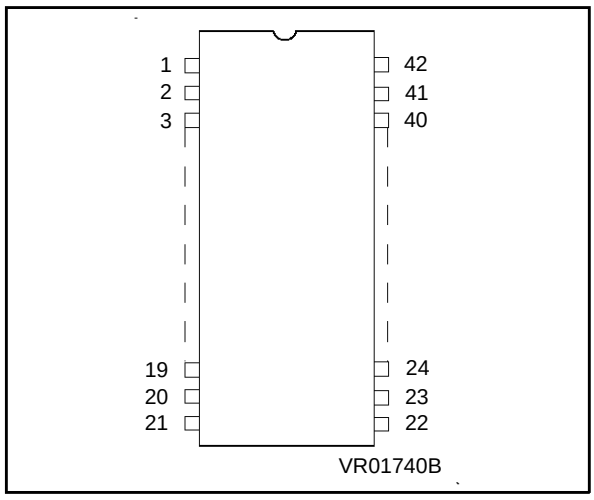
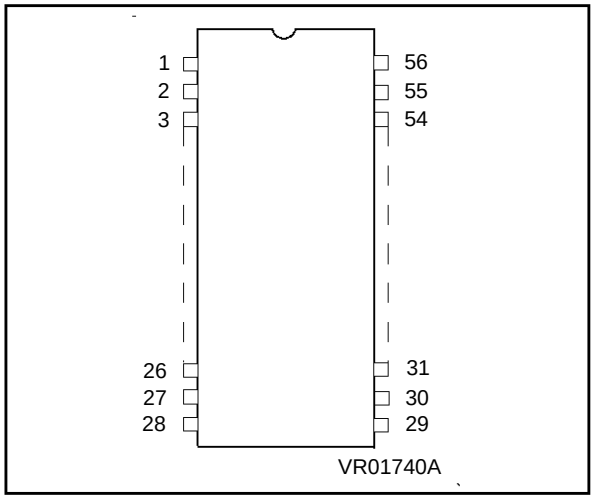


Figure 2. 56 Pin Shrink DIP Pinout



ST9291J Pin Description

Pin	Pin name	Pin	Pin name
1	P2.0/INT7	42	P2.1/INT5/AIN1
2	$\overline{\text{RESET}}$	41	P2.2/INT0/AIN2
3	P0.7	40	P2.3/INT6/VSO1
4	P0.6	39	P2.4/NMI
5	P0.5	38	P2.5/AIN3/VSO2
6	P0.4	37	OSCIN
7	P0.3	36	OSCOUT
8	P0.2	35	P4.7/PWM7/ EXTRG (AD)
9	P0.1	34	P4.6/PWM6
10	P0.0	33	P4.5/PWM5
11	P3.7	32	P4.4/PWM4
12	P3.6	31	P4.3/PWM3
13	P3.5	30	P4.2/PWM2
14	P3.4	29	P4.1/PWM1
15	P3.3/B	28	P4.0/PWM0
16	P3.2/G	27	VSYNC
17	P3.1/R	26	HSYNC
18	P3.0/FB	25	AV_{DD}
19	P5.1/SDIO	24	PLLR
20	P5.0/SCK/INT2	23	PLLF
21	V_{DD}	22	V_{SS}

Notes (N Package only) :

1. N.C. means “not connected”
2. Pins 14 and 28 (V_{DD}) are internally connected

ST9291N Pin Description

Pin	Pin name	Pin	Pin name
1	P2.1/INT5/AIN1	56	P2.2/INT0/AIN2
2	P2.0/INT7	55	P2.3/INT6/VSO1
3	$\overline{\text{RESET}}$	54	P2.4/NMI
4	P0.7	53	P2.5/AIN3/VSO2
5	P0.6	52	P1.0
6	P0.5	51	P1.1
7	N.C.(1)	50	P1.2
8	P0.4	49	P1.3
9	P0.3	48	P1.4
10	P0.2	47	P1.5
11	P0.1	46	P1.6
12	P0.0	45	P1.7
13	N.C.(1)	44	OSCIN
14	V_{DD}(2)	43	OSCOUT
15	N.C.(1)	42	P4.7/PWM7/ EXTRG (AD)
16	P3.7	41	P4.6/PWM6
17	P3.6	40	P4.5/PWM5
18	P3.5	39	P4.4/PWM4
19	P3.4	38	P4.3/PWM3
20	P3.3/B	37	P4.2/PWM2
21	P3.2/G	36	P4.1/PWM1
22	P3.1/R	35	P4.0/PWM0
23	P3.0/FB	34	VSYNC
24	P5.3	33	HSYNC
25	P5.2	32	AV_{DD}
26	P5.1/SDIO	31	PLLR
27	P5.0/SCK/INT2	30	PLLF
28	V_{DD}(2)	29	V_{SS}

GENERAL DESCRIPTION

The ST9291 is a ROM member of the ST9 family of microcontrollers, completely developed and produced by SGS-THOMSON Microelectronics using a proprietary n-well HCMOS process.

The ROM parts are fully compatible with their EPROM and OTP (One-Time Programmable) versions, which may be used for the prototyping and pre-production phases of development.

The nucleus of the ST9291 is the advanced ST9 Core which includes the Central Processing Unit (CPU), the Register File, a 16-bit Timer/Watchdog with 8-bit Prescaler, a Serial Peripheral Interface supporting S-bus, I^2C-bus and IM-bus Interface, plus two 8-bit I/O ports. The Core has independent memory and register buses allowing a high degree of pipelining to add to the efficiency of the code execution speed of the extensive instruction set.

The powerful I/O capabilities demanded by microcontroller applications are fulfilled by the ST9291 with up to 32/42 I/O lines dedicated to digital Input/Output. These lines are grouped into up to six I/O Ports and can be configured on a bit basis under software control to provide timing, status signals, timer inputs and outputs, analog inputs, external interrupts, OSD (On Screen Display) output and serial or parallel I/O.

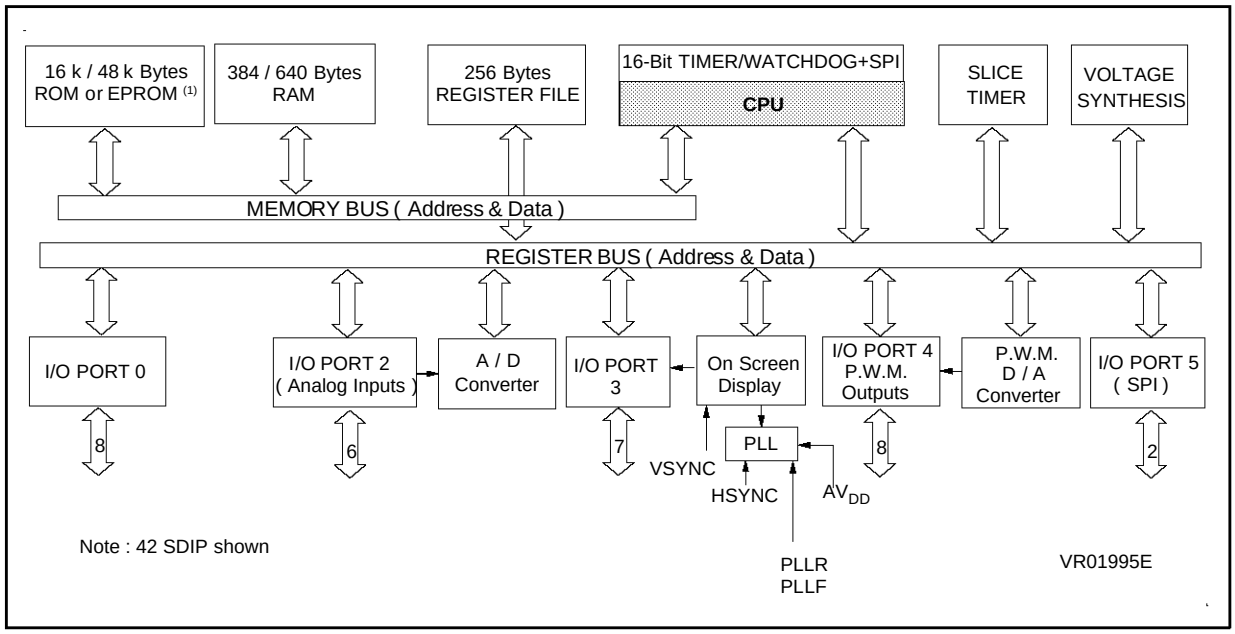
Three basic memory spaces are available to support this wide range of configurations: Program Memory, Data Memory and the Register File, which includes the control and status registers of the on-chip peripherals.

The human interface is provided by the On Screen Display module, this can produce up to 15 lines of up to 34 characters from a ROM defined 128 character set. The 9x13 character can be modified by 4 different pixel sizes, with character rounding, and formed into words with colour and format attributes.

A 14-bit VS (Voltage Synthesis) output using the PWM (Pulse Width Modulation)/BRM (Bit Rate Modulation) is present to generate tuning voltages for low-mid range TV set applications. The tuning voltage is output on one of two separate output pins.

A 16-bit Slice Timer with an 8-bit Prescaler is also present.

Figure 3. ST9291 Block Diagram



Note 1. EPROM version only

GENERAL DESCRIPTION (Continued)

The control of TV or Satellite receiver setting can be done by up to eight 8-bit PWM outputs, with a frequency maximum of 23,437Hz at 8-bit resolution (INTCLK = 12MHz). Low resolutions with higher frequency operation can be programmed.

In addition there is a 3 channel Analog to Digital Converter with integral sample and hold, fast 5.75µs conversion time and 6-bit guaranteed resolution.

PIN DESCRIPTION

VSYNC. *Vertical Sync.* Vertical video synchronisation input to OSD. Positive or negative polarity.

HSYNC. *Horizontal Sync.* Horizontal video synchronisation input to OSD. Positive or negative polarity.

PLLF. *PLL Filter input.* Filter input for the OSD for PLL feed-back.

PLLR. *PLL Resistor connection pin.* For resistor connection to select the PLL gain adjust.

$\overline{\text{RESET}}$. *Reset (input, active low).* The ST9 is initialised by the Reset signal. With the deactivation of $\overline{\text{RESET}}$, program execution begins from the Program memory location pointed to by the vector contained in program memory locations 00h and 01h.

OSCIN, OSCOUT. *Oscillator (input and output).* These pins connect a parallel-resonant crystal (24MHz maximum), or an external source to the on-chip clock oscillator and buffer. OSCIN is the input of the oscillator inverter and internal clock generator; OSCOUT is the output of the oscillator inverter.

AV_{DD}. Analog V_{DD} of PLL. This pin must be tied to V_{DD} externally to the ST9291.

V_{DD}. Main Power Supply Voltage (5V±10%)

V_{SS}. Digital Circuit Ground.

P0.0-P0.7, P2.0-P2.5, P3.0-P3.7, P4.0-P4.7, P5.0-P5.1 (J suffix)
P0.0-P0.7, P1.0-P1.7, P2.0-P2.5, P3.0-P3.7, P4.0-P4.7, P5.0-P5.3 (N suffix) *I/O Port Lines (Input/Output, TTL or CMOS compatible).* 32/42 lines grouped into I/O ports, bit programmable under program control as general purpose I/O or as Alternate functions (see next section).

P4.0 - P4.7 are high voltage (12V) open drain outputs. The voltage in open drain output mode for all other I/O bits must not exceed V_{DD}.

I/O Port Alternate Functions.

Each pin of the I/O ports of the ST9291 may assume software programmable Alternative Functions as shown in the Pin Configuration Drawings. Table 1 shows the Functions allocated to each I/O Port pin.

PIN DESCRIPTION (Continued)

Table 1.ST9291 I/O Port Alternative Function Summary

I/O PORT	Name	Function	Alternate Function	Pin Assignment	
Port.bit				9291J	9291N
P0.0		I/O		10	12
P0.1		I/O		9	11
P0.2		I/O		8	10
P0.3		I/O		7	9
P0.4		I/O		6	8
P0.5		I/O		5	6
P0.6		I/O		4	5
P0.7		I/O		3	4
P1.0		I/O		-	52
P1.1		I/O		-	51
P1.2		I/O		-	50
P1.3		I/O		-	49
P1.4		I/O		-	48
P1.5		I/O		-	47
P1.6		I/O		-	46
P1.7		I/O		-	45
P2.0	INT7	I	External Interrupt 7 with Schmitt Trigger	1	2
P2.1	INT5	I	External Interrupt 5 with Schmitt Trigger	42	1
P2.1	AIN1	I	A/D Analog Input 1	42	1
P2.2	INT0	I	External Interrupt 0	41	56
P2.2	AIN2	I	A/D Analog Input 2	41	56
P2.3	INT6	I	External Interrupt 6	40	55
P2.3	VSO1	O	Voltage Synthesis Output 1	40	55
P2.4	NMI	I	Non-Maskable Interrupt	39	54
P2.5	AIN3	I	A/D Analog Input 3	38	53
P2.5	VSO2	O	Voltage Synthesis Output 2	38	53
P3.0	FB	O	Fast Blanking OSD output	18	23
P3.1	R	O	Red Video Colour OSD output	17	22
P3.2	G	O	Green Video Colour OSD output	16	21
P3.3	B	O	Blue Video Colour OSD output	15	20

PIN DESCRIPTION (Continued)

Table 1. ST9291 I/O Port Alternative Function Summary (Continued)

I/O PORT	Name	Function	Alternate Function	Pin Assignment	
Port.bit				9291J	9291N
P3.4		I/O		14	19
P3.5		I/O		13	18
P3.6		I/O		12	17
P3.7		I/O		-	16
P4.0	PWM0	O	PWM Output 0	28	35
P4.1	PWM1	O	PWM Output 1	29	36
P4.2	PWM2	O	PWM Output 2	30	37
P4.3	PWM3	O	PWM Output 3	31	38
P4.4	PWM4	O	PWM Output 4	32	39
P4.5	PWM5	O	PWM Output 5	33	40
P4.6	PWM6	O	PWM Output 6	34	41
P4.7	PWM7	O	PWM Output 7	35	42
P4.7	EXTRG	I	A/D External Trigger	35	42
P5.0	SCK	O	SPI Serial Clock (1)	20	27
P5.0	INT2	I	External Interrupt 2 (1)	20	27
P5.1	SDIO	I/O	SPI Serial Data Input/Output (1)	19	26
P5.2		I/O		-	25
P5.3		I/O		-	24

Notes.

1. The alternate functions of SCK/INT2 and SDIO may be swapped by using the SWAP Register Function.
2. Schmitt trigger options are available as a mask option for any input pin.

1 CORE DESCRIPTION

1.1 CORE ARCHITECTURE

1.1.1 INTRODUCTION

The Core or Central Processing Unit (CPU) of the ST9 includes the 8 bit Arithmetic Logic Unit and the 16 bit Program Counter, System and User Stack Pointers. The microcoded Instruction Set is highly optimised for both byte (8 bit) and word (16 bit) data, BCD and Boolean data types, with 14 addressing modes.
Three independent buses are controlled by the Core, a 16 bit Memory bus, an 8 bit Register addressing bus and a 6 bit Interrupt/DMA bus connected to the interrupt and DMA controllers in the on-chip peripherals and the Core. This multiple bus architecture allows a high degree of pipelining and parallel operation, giving the ST9 its efficiency in both numerical calculations and communication with the on-chip peripherals.

1.1.2 ADDRESS SPACES

The ST9 has three separate address spaces:

- Register File: 240 8-bit registers plus up to 64 pages of 16 bytes each, located in the on-chip peripherals.
- Data memory with up to 64K (65536) bytes
- Program memory with up to 64K (65536) bytes

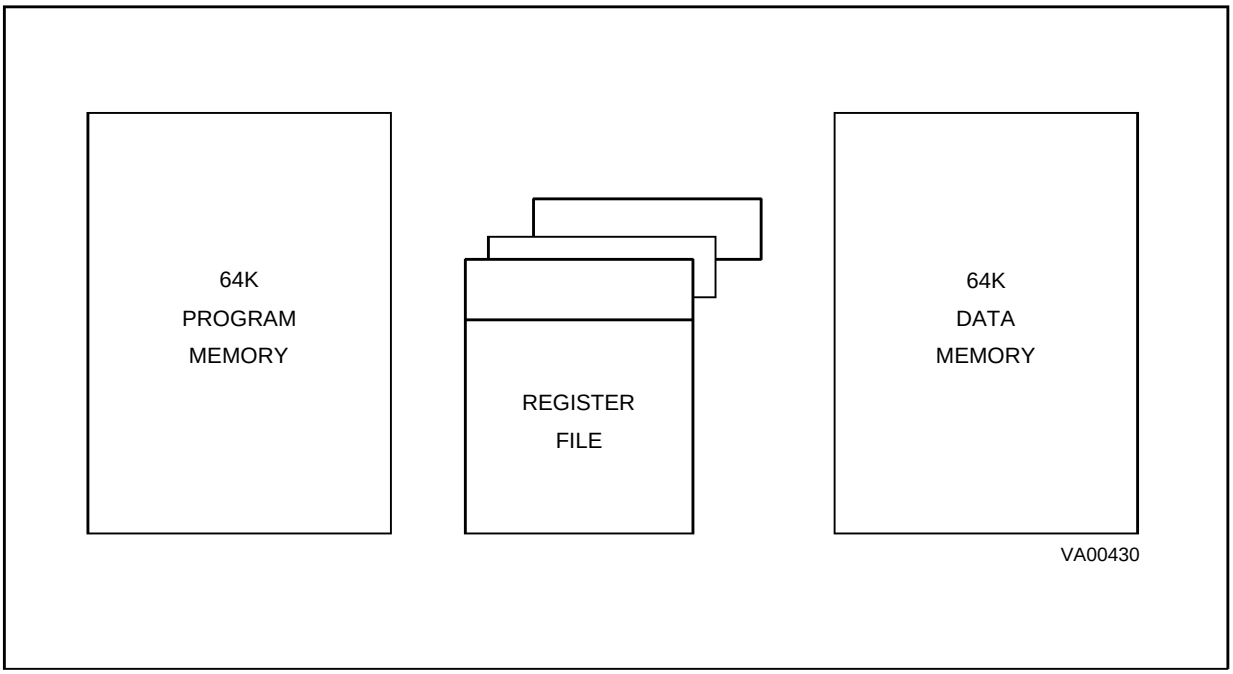
The Data and Program memory spaces will be addressed in further detail in section 1.3.

1.1.2.1 Register File

The Register File consists of:

- 224 general purpose registers R0 to R223
- 16 system registers in the System Group (R224 to R239).
- I/O pages depending on the configuration of the ST9, each containing up to 16 registers, with paging facilities based on the top group (R240 to R255).

Figure 1-4. Address Spaces



ADDRESS SPACES (Continued)

Figure 1-5. Register Grouping

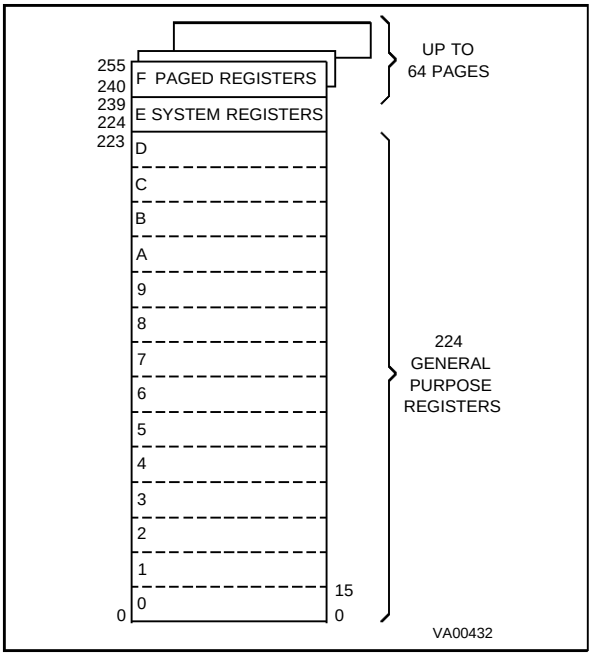


Figure 1-6. Page Pointer Configuration

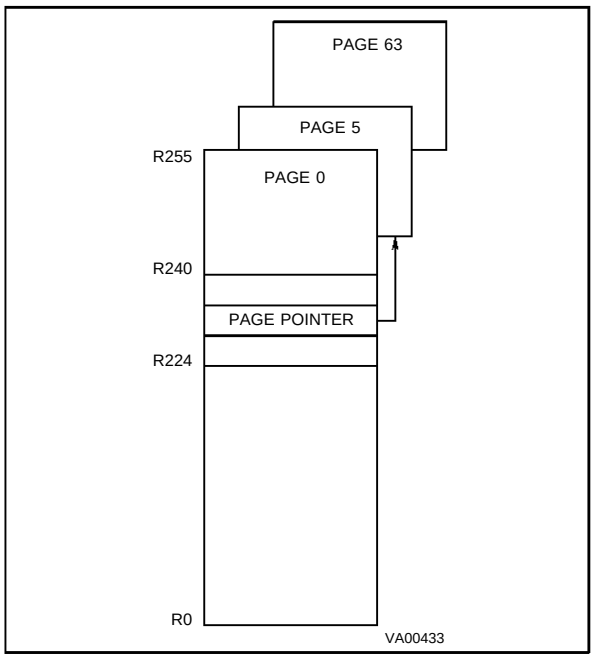
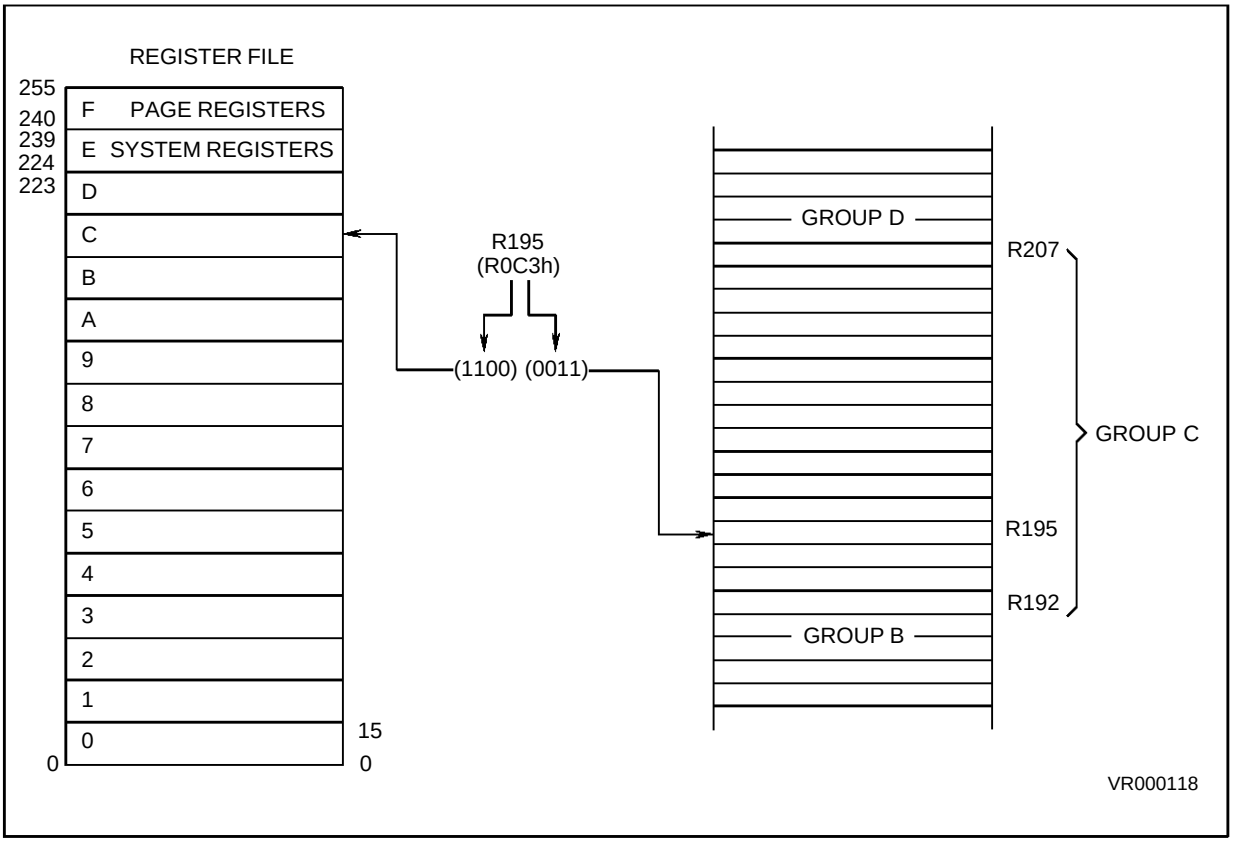


Figure 1-7. Addressing the Register File



ADDRESS SPACES (Continued)

1.1.2.2 Addressing Registers

All registers in the Register File and pages can be specified by using a decimal, hex or binary address, e.g. R231, RE7h or R11100111b is the same register.
The registers can be referred to by their hexadecimal group address, so that registers R0-R15 form group 0, R160-R175 form group A and so on.

Working Register Addresses

The 8-bit register address is formed by 2 nibbles, for example, for register R195 or RC3h or R11000011, 1100 specifies the 13th group (i.e. group C) and 0011 specifies the 3rd register in that group.

Working registers are addressed by supplying the least significant nibble in the instruction and adding it to the most significant nibble found in the Register Pointer (R233). Working register addressing is shown in Figure 1-7.

System Registers

The 16 system registers at addresses R224 to R239 form Group E.
The system registers are addressable using any of the 4 register addressing modes and the most significant nibble will, in all cases, be 14 (0Eh).

Paged Registers

There are a maximum of 64 pages each containing 16 registers. These are addressed using the register addressing modes with the addition of the Page Pointer register, R234. This register selects the page to be addressed in group F and once set, does not need to be changed if two or more registers on the same page are to be addressed in succession.

Therefore if the Page Pointer, R234, is set to 5, the instructions

```
spp 5
ld R242, r4
```

will load the contents of working register r4 into the third register (R242) of page 5.

These paged registers hold data and control registers related to the on-chip peripherals, and thus the configuration depends upon the peripheral organisation of each ST9 family member. i.e. pages only exist if the peripheral exists.

Available pages are shown in Table 1-3.

1.1.2.3 Input/Output Ports

The Input/Output ports are located in two areas. The port registers for Ports 0-5 are located at the bottom of the System register group in locations R224 to R229.

Each Port has three associated Control registers, which determine the individual pin modes (I/O, Open-Drain etc). These registers are located in pages 2 and 3.

Table 1-2. Register File Organization

Hex. Address	Decimal Address	Function	Register File Group
F0-FF	240-255	Paged Registers	Group F
E0-EF	224-239	System Registers	Group E
D0-DF	208-223	General Purpose Registers	Group D
C0-CF	192-207		Group C
B0-BF	176-191		Group B
A0-AF	160-175		Group A
90-9F	144-159		Group 9
80-8F	128-143		Group 8
70-7F	112-127		Group 7
60-6F	96-111		Group 6
50-5F	80-95		Group 5
40-4F	64-79		Group 4
30-3F	48-63		Group 3
20-2F	32-47		Group 2
10-1F	16-31		Group 1
00-0F	00-15		Group 0

ADDRESS SPACES (Continued)

Table 1-3. Group F Peripheral Organization

Applicable for ST9291

DEC	HEX	00 00	02 02	03 03	0B 11	28 40	29 41	2A 42	3B 59	3E 62	
R255	RFF	SWAP	RESER						VSO		RFF
R254	RFE	SPI						RESER			RFE
R253	RFD		PORT 3						RESER		RFD
R252	RFC	WCR									RFC
R251	RFB		RESER	RESER							RFB
R250	RFA	T/WD			RESER						RFA
R249	RF9		PORT 2			OSD	OSD			RESER	RF9
R248	RF8					CHAR	CHAR	OSD			RF8
R247	RF7		RESER			1 to 16	17 to 32		PWM		RF7
R246	RF6										RF6
R245	RF5	EXT INT	PORT1	PORT 5							RF5
R244	RF4										RF4
R243	RF3		RESER	RESER				OSD			RF3
R242	RF2				SLICE			CHAR			RF2
R241	RF1	RESER	PORT 0	PORT 4	TIMER			33 to 36		A/D	RF1
R240	RF0									CONV	RF0

1.1.3 SYSTEM REGISTERS

Following is the description of System Registers. For PORT0 to PORT5 Registers, please refer to I/O Port Chapter.

Figure 1-8. System Register

R239 (EFh)	SYS. STACK POINTER LOW
R238 (EEh)	SYS. STACK POINTER HIGH
R237 (EDh)	USER STACK POINTER LOW
R236 (ECh)	USER STACK POINTER HIGH
R235 (EBh)	MODE REGISTER
R234 (EAh)	PAGE POINTER
R233 (E9h)	REGISTER POINTER 1
R232 (E8h)	REGISTER POINTER 0
R231 (E7h)	FLAGS
R230 (E6h)	CENTRAL INT. CNTL REG
R229 (E5h)	PORT5
R228 (E4h)	PORT4
R227 (E3h)	PORT3
R226 (E2h)	PORT2
R225 (E1h)	PORT1
R224 (E0h)	PORT0

1.1.3.1 Central Interrupt Control Register

This Register CICR is located in the system Register Group at the address R230 (E6h). Please refer to “INTERRUPT” and “DMA” chapters in order to get the background of the ST9 interrupt philosophy.

CICR R230 (E6h) System Read/Write
Central Interrupt Control Register
Reset Value : 1000 0111

7							0
GCEN	TLIP	TLI	IEN	IAM	CPL2	CPL1	CPL0

b7 = **GCEN:** *Global Counter Enable*. This bit is the Global Counter Enable of the Multifunction Timers. The GCEN bit is ANDed with the CE (Counter Enable) bit of the Timer Control Register (explained in the Timer chapter) in order to enable the Timers when both bits are set. This bit is set after the Reset cycle.

b6 = **TLIP:** *Top Level Interrupt Pending*. This bit is automatically set when a Top Level Interrupt Request is recognized. This bit can also be set by Software in order to simulate a Top Level Interrupt Request.

b5 = **TLI:** *Top Level Interrrupt bit*. When this bit is set, a Top Level interrupt request is acknowledged depending on the IEN bit and the TLNM bit (in Nested Interrupt Control Register). If the TLM bit is reset the top level interrupt acknowledgement depends on the TLNM alone.

b4 = **IEN:** *Enable Interrupt*. This bit, (when set), allows interrupts to be accepted. When reset no interrupts other than the NMI can be acknowledged. It is cleared by interrupt acknowledgement for concurrent mode and set by interrupt return (`iret`). It can be managed by hardware and software (`ei` and `di` instruction).

b3 = **IAM:** *Interrupt Arbitration Mode*. This bit covers the selection of the two arbitration modes, the Concurrent Mode being indicated by the value “0” and the Fully Automatic Nested Mode by the value “1”. This bit is under software control.

b2-b0 = **CPL2-CPL0:** *Current Priority Level*. These three bits record the priority level of the interrupt presently under service (i.e. the Current Priority Level, CPL). For these priority levels 000 is the highest priority and 111 is the lowest priority. The CPL bits can be set by hardware or software and give the reference by which following interrupts are either left pending or able to interrupt the current interrupt. When the present interrupt is replaced by one of a greater priority, the current priority value is automatically stored until required.

SYSTEM REGISTERS (Continued)

1.1.3.2 Flag Register

The Flag Register contains 8 flags indicating the status of the ST9. During an interrupt the flag register is automatically stored in the system stack area and recalled at the end of the interrupt service routine so that the ST9 is returned to the original status. This occurs for all interrupts and, when operating in the nested mode, up to seven versions of the flag register may be stored.

FLAGR R231 (E7h) System Read/Write Flag Register
Reset value: **undefined**

7							0
C	Z	S	V	DA	H	UF	DP

b7 = **C:** *Carry Flag*. The carry flag C is affected by the following instructions:

Addition (`add, addw, adc, adcw`),
Subtraction (`sub, subw, sbc, sbcw`),
Compare (`cp, cpw`),
Shift Right Arithmetic (`sra, sraw`),
Rotate (`rrc, rrcw, rlc, rlcw, ror, rol`),
Decimal Adjust (`da`),
Multiply and Divide (`mul, div, divws`).

When set, it generally indicates a carry out of the most significant bit position of the register being used as an accumulator (bit 7 for byte and bit 15 for word operations).

The carry flag can be set by the Set Carry Flag (`scf`) instruction, cleared by the Reset Carry Flag (`rcf`) instruction, and complemented (changed to "0" if "1", and vice versa) by the Complement Carry Flag (`ccf`) instruction.

b6 = **Z:** *Zero Flag*. The Zero flag is affected by the following instructions:

Addition (`add, addw, adc, adcw`),
Subtraction (`sub, subw, sbc, sbcw`),
Compare (`cp, cpw`),
Shift Right Arithmetic (`sra, sraw`),
Rotate (`rrc, rrcw, rlc, rlcw, ror, rol`),
Decimal Adjust (`da`),
Multiply and Divide (`mul, div, divws`),
Logical (`and, andw, or, orw, xor, xorw, cpl`),
Increment and Decrement (`inc, incw, dec, decw`),
Test (`tm, tmw, tcm, tcmw, btset`).

In most cases, the Zero flag is set when the register being used as an accumulator register is zero, following one of the above operations.

b5 = **S:** *Sign Flag*. The Sign flag is affected by the same instructions as the Zero flag.

The Sign flag is set when bit 7 (for byte operation) or bit 15 (for word operation) of the register used as an accumulator is one.

b4 = **V:** *Overflow Flag*. The Overflow flag is affected by the same instructions as the Zero and Sign flags.

When set, the Overflow flag indicates that a two's-complement number, in a result register, is in error, since it has exceeded the largest (or is less than the smallest), number that can be represented in twos-complement notation.

b3 = **DA:** *Decimal Adjust Flag*. The Decimal Adjust flag is used for BCD arithmetic. Since the algorithm for correcting BCD operations is different for addition and subtraction, this flag is used to specify which type of instruction was executed last, so that the subsequent Decimal Adjust (da) operation can perform its function correctly.
The Decimal Adjust flag cannot normally be used as a test condition by the programmer.

b2 = **H:** *Half Carry Flag*. The Half Carry flag indicates a carry out of (or a borrow into) bit 3, as the result of adding or subtracting two 8-bit bytes, each representing two BCD digits. The Half Carry flag is used by the Decimal Adjust (da) instruction to convert the binary result of a previous addition or subtraction into the correct BCD result.
Like the Decimal Adjust flag, this flag is not normally accessed by the user.

b1 = **UF:** *User Flag*. Bit 1 in the flag register (UF) is available to the user, but it must be set or cleared by an instruction.

b0 = **DP:** *Data/Program Memory Flag*. This bit in the flag register indicates which memory area is addressed. Its value is affected by the Set Data Memory (`sdm`) and Set Program Memory (`spm`) instructions.

If the bit is set, the ST9 addresses the Data Memory Area; when the bit is cleared, the ST9 addresses the Program Memory Area. By reading this bit, the user can verify in which memory area the processor is working. The user writes this bit with the `sdm` or `spm` instructions.

SYSTEM REGISTERS (Continued)

1.1.3.3 Register Pointing Techniques

Two registers, R232 and R233, within the system register group, are available for register pointing. R232 and R233 may be used together as a single pointer for a 16 register working space or separately for two 8 register spaces, in which case R232 becomes Register Pointer 0 (RP0) and R233 becomes Register Pointer 1 (RP1).

The instructions `srp, srp0` and `srp1` (the Set Register Pointer instructions) automatically inform the ST9 whether the Register File is to operate with a single 16-register group or two 8-register groups. The `srp0` and `srp1` instructions automatically set the twin 8-register group mode while the `srp` instruction sets the single 16-register group mode. There is no limitation on the order or positions of these chosen register groups other than they must be on 8 or 16 register boundaries.

The addressing of working registers involves use of the Register Pointer value plus an offset value given by the number of the addressed working register.

When addressing a register, the most significant nibble (bits 4-7) gives the group address and the least significant nibble (bits 0-3) gives the register within that group.

REGISTER POINTER 0

RP0 R232 (E8h) System Read/Write
Register Pointer 0
Reset Value : **undefined**

7							0
RG7	RG6	RG5	RG4	RG3	RPS	D1	D0

b7-b3 = **RG7-RG3:** *Register Group number*. These bits contain the number (from 0 to 31) of the group of working registers indicated in the instructions `srp0` or `srp`. When using a 16-register group, a number between 0 and 31 must be used in the srp instruction indicating one of the two adjacent 8-register group of working registers used. RG7 is the MSB.

b2 = **RPS:** *Register Pointer Selector*. This bit is set by the instructions `srp0` and `srp1` to indicate that a double register pointing mode is used. Otherwise, the instruction `srp` resets the RPS bit to zero to indicate that a single register pointing mode is used.

b1,b0 = **D1,D0:** These bits are fixed by hardware to zero and are not affected by any writing instruction trying to modify their value.

REGISTER POINTER 1

RP1 R233 (E9h) System Read/Write
Register Pointer 1
Reset Value : **undefined**

7							0
RG7	RG6	RG5	RG4	RG3	RPS	D1	D0

This register is used only with double register pointing mode; otherwise, using single register pointing mode, the RP1R register has to be considered as reserved and not usable as a general purpose register.

b7-b3 = **RG7-RG3:** *Register Group number*. These bits contain the number (from 0 to 31) of the group of 8 working registers indicated in the instructions srp1. Bit 7 is the MSB.

b2 = **RPS:** *Register Pointer Selector*. This bit is automatically set by the instructions `srp0` and `srp1` to indicate that a double register pointing mode is used. Otherwise the instruction srp reset the RPS bit to zero to indicate that a single register pointing mode is used.

b1,b0 = **D1,D0:** These bits are hardware fixed to zero and are not affected by any writing instruction trying to modify their value.

Note. If working in twin 8-register group mode but only using `srp0` (i.e. only using one 8-register group) the unused register (R233) is to be considered as reserved and not usable as a general purpose register.

The group of registers immediately below the system registers (i.e. group D, R208-R223) can only be accessed via the Register Pointers. To address group D then, it is necessary to set the Register Pointer to group D and then use the addressing procedure for working registers. The programmer is required to remember that the group D should be used as a stacking area. This point is also covered in the Stack Pointers paragraph.

SYSTEM REGISTERS (Continued)

EXAMPLES

Using the Single 16 Register Group

When the system is operating in the single 16-register group mode, the registers are referred to as r0-r15. In this mode, the offset value (i.e. the number of the working register referred to) is supplied in the address (preceded by a small r, e.g. r5) and is added to the Register Pointer 0 value to give the absolute address.

For example, if the Register Pointer contains the value 70h, then working register r7 would have the absolute address, R77h.

In this mode, the single 16-registers group will always start from the lowest even number equal or lower to the number given in the instruction.

Example: `srp #3` is equivalent to `srp #2`.

Using the Twin 8-Register Group

When working in the twin working group mode, the registers pointed by Register Pointer 0 (RP0R), are referred as r0-r7 and those pointed by Register Pointer 1 (RP1R), are referred to as r8-r15, regardless of their absolute addresses. In this mode, when operating with the first 8 working registers (i.e. r0 - r7) the working register number acts as an offset which is added to the value in Register Pointer 0.

So if Register Pointer 0 contains the value 96, then working register 0 has the absolute address 96, working register 5 has the absolute address 101, and so on. The second group of working registers, r8-r15, has the offset values 0 to 7 respectively (i.e. r8 has the offset value 0, r9 has the offset value 1, and so on), this offset value being added to the value in Register Pointer 1.

For example, given that the value in Register Pointer 1 is 32, then working register 12 supplies an offset value of 4 (given by 12 minus 8) to the value in Register Pointer 1 to give an absolute address of 36.

Figure 1-9. Single 16 Register pointing Mode

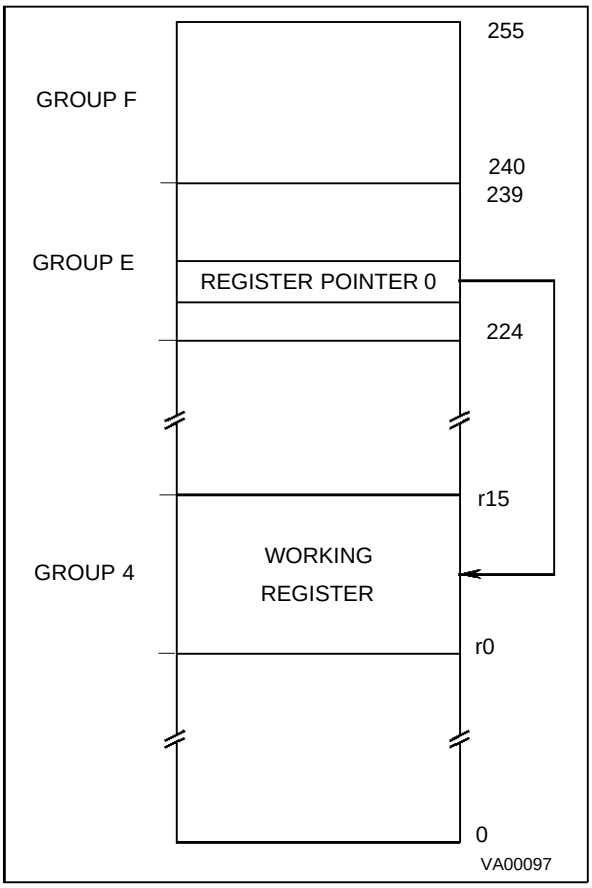
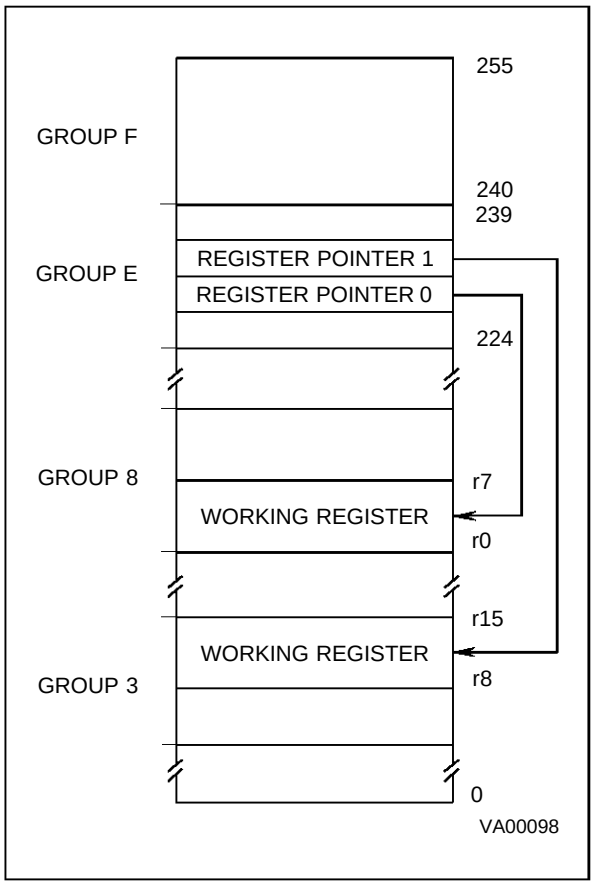


Figure 1-10. Double Register pointing Mode



SYSTEM REGISTERS (Continued)

1.1.3.4 Page Configuration

The pages are available to be used for the storage of control information (such as interrupt vector pointers) relevant to particular peripherals. There are up to 64 pages (each with 16 registers) based on registers R240-R255. These paged registers are addressable via the page pointer register (PPR), which is system register R234.

To address a paged register the page pointer register (R234) must be loaded with the relevant page number using the spp instruction (Set Page Pointer) and subsequently any address from the top (F) group (R240-R255) will be referred to that page.

For example if register 23 contains the value 44, the following sequence loads the third register R242 on page 5 with the value 44.

```
spp 5
ld R242, R23
```

PPR R234 (EAh) System Read/Write Page Pointer Register
Reset value **: undefined**

7							0
PP7	PP6	PP5	PP4	PP3	PP2	D1	D0

b7-b2 = **PP7-PP2:** *Page Pointer*. These bits contain the number (between 0 to 63) of the page chosen by the instruction ssp (Set Page Pointer). PP7 is the MSB of the page address. Once the page pointer has been set, there is no need to refresh it unless a different page is required.

b1-b0 = **D1,D0:** These bits are fixed by hardware to zero and are not affected by any writing instruction trying to modify their value.

PAGE 0 contains the control registers of:

- the external interrupt
- the watchdog timer
- the wait logic states
- the serial peripheral interface (SPI)

1.1.3.5 Mode Registers

This register MODER is located in the System Register Group at the address 235.

Using this register it is possible:

- to select either internal or external System and User Stack area,
- to manage the clock frequency
- to enable the Bus request and Wait signals when interfacing external memory.

MODER R235 (EBh) System Read/Write Mode Register
Reset value : 1110 0000

7							0
SSP	USP	DIV2	PRS2	PRS1	PRS0	BRQEN	HIMP

b7 = **SSP:** *System Stack Pointer*. This bit selects internal (in the Register File) or external (in the external Data Memory) System Stack area, logical “1” for internal, and logical “0” for external. After Reset the value of this bit is “1”.

b6 = **USP:** *User Stack Pointer*. Same as bit 7 for the User Stack Pointer;

b5 = **DIV2:** *OSCIN Clock Divided by 2*. This bit controls the divide by 2 circuit which operates on the OSCIN Clock. A logical “1” value means that the OSCIN clock is internally divided by 2, and a logical “0” value means that no division of the OSCIN Clock occurs.

b4-b2 = **PRS2-PRS0:** *ST9 CPUCLK Prescaler*. These bits load the prescaling module of the internal clock (INTCLK). The prescaling value selects the frequency of the ST9 clock, which can be divided by 1 to 8. See Clock chapter for more information.

b1 = **BRQEN:** *Bus Request Enable*. This bit must be held to “0”.

b0 = **HIMP:** *High Impedance Enable*. This bit must be held to “0”.

SYSTEM REGISTERS (Continued)

1.1.3.6 Stack Pointers

There are two separate, double register stack pointers available (named System Stack Pointer and User Stack Pointer), both of which can address registers or memory.

The stack pointers point to the bottom of the stacks which are filled using the push commands and emptied using the pop commands. The stack pointer is automatically pre-decremented when data is "pushed in" and post-incremented when data is "popped out".

For example, the register address space is selected for a stack and the corresponding stack pointer register contains 220. When a byte of data is "pushed" into the stack, the stack pointer register is decremented to 219, then the data byte is "loaded" into register 219. Conversely, if a stack pointer register contains 189 and a byte of data is "popped" out, the byte of data is then extracted from the stack and then the stack pointer register is incremented to 190.

The push and pop commands used to manage the system stack area are made applicable to the user stack by adding the suffix U, while to use a stack instruction for a word a W is added.

For example push inserts data into the system stack, but an added U indicates the user stack and W means a word, so the instruction pushuw loads a word into the bottom of the user stack.

If the User Stack Pointer register contains 223 (working in register space) the instruction pushuw will decrement User Stack Pointer register to 222 and then load a word into register R222 and R221.

When bytes (or words) are "popped out" the values in those registers are left unchanged until fresh data is loaded into those locations. Thus when data is "popped" out from a stack area, the stack content remains unchanged.

Note. Stacks must not be located in the pages or the system register area.

The System Stack area and The System Stack Pointer

The System Stack area is used for the storage of temporarily suspended system and/or control registers, i.e. the Flag register and the Program counter, while interrupts are being serviced. For subroutine execution only the Program Counter needs to be saved in the System stack area.

There are two situations when this occurs automatically, one being when an interrupt occurs and the other when the instruction call subroutine is used. When the system stack area is in the Register File, the stack pointer, which points to the bottom of the stack, only needs one byte for addressing, in which case the System Stack Pointer Low Register (R239) is sufficient for addressing purposes. As a result the System Stack Pointer High Register (R238) becomes redundant BUT must be considered as reserved (please refer also to "spurious" memory access section). Clearly when the stack is external a full word address is necessary and so both registers are used to point, the even register providing the MSB and the odd register providing the LSB.

The User Stack area and User Stack Pointer

The User Stack area is completely free from all interference from automatic operations and so it provides a totally user controlled stacking area, that area being in any part of the memory which is of a RAM nature, or the first 14 groups of the general Register File i.e. not in the System register or Paged group.

The User Stack Pointer consists of two registers, R236 and R237, which are both used for addressing an external stack, while, when stacking in the Register File, the User Stack Pointer High Register, R236, becomes redundant but must be considered as reserved.

SYSTEM REGISTERS (Continued)

Stack location

Care is necessary when managing stacks as there is no limit to stack sizes apart from the bottom of any address space in which the stack is placed. Consequently programmers are advised to use a stack pointer value as high as possible, particularly when using the Register File as a stacking area. This will also benefit programmers who may locate the stacks in group D using, for example the instruction `ld R237, #223` which loads the value 223 into the User Stack Pointer Low Register. The Programmer will not need to remember to set the Register Pointer to 208 to gain access to registers in the D-group, a problem outlined in Register Pointing Techniques paragraph.

Stacks may be located anywhere in the first 14 groups of the Register File (internal stacks) or the data memory (external stacks). It is not necessary to set the data memory using the instruction sdm as external stack instructions automatically use the data memory.

Figure 1-11. System and/or User Stack in Register Stack Mode

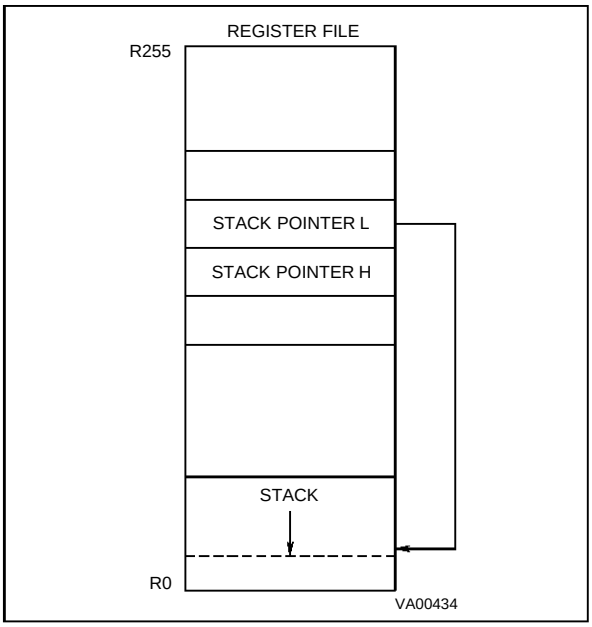
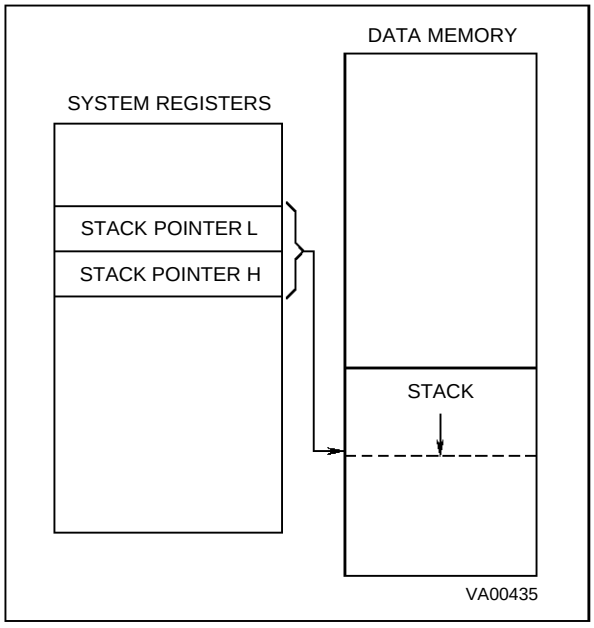


Figure 1-12. System and/or User Stack in Memory Stack Mode



USP R236 (ECh) System Read/Write
User Stack Pointer High Byte
Reset value: **undefined**

7							0
USP15	USP14	USP13	USP12	USP11	USP10	USP9	USP8

USP R237 (EDh) System Read/Write
User Stack Pointer Low Byte
Reset value: **undefined**

7							0
USP7	USP6	USP5	USP4	USP3	USP2	USP1	USP0

SSP R238 (EEh) System Read/Write
System Stack Pointer High Byte
Reset value: **undefined**

7							0
SSP15	SSP14	SSP13	SSP12	SSP11	SSP10	SSP9	SSP8

SSP R239 (EFh) System Read/Write
System Stack Pointer Low Byte
Reset value: **undefined**

7							0
SSP7	SSP6	SSP5	SSP4	SSP3	SSP2	SSP1	SSP0

1.2 MEMORY

1.2.1 INTRODUCTION

The memory of the ST9291 is functionally divided into two areas, the Register File and Memory. The Memory may optionally be divided into two spaces, Program Memory for Program code and Data Memory for Data.

The memory spaces are selected by the execution of the SDM and SPM instructions (Set Data Memory and Set Program Memory, respectively). There is no need to use either of these instructions again until the memory area required is to be changed.

1.2.1.1 Program Space

The Program memory space of the ST9291 consists of 48K bytes of on-chip ROM (addressed from 0 to BFFF) and 640 bytes of on-chip RAM (addressed from FD80h to FFFFh); refer to the memory map tables and drawing on the following page for the memory mapping for other ROM sizes. The first 256 memory locations from address 0 to 00FFh (hexadecimal) hold the Reset Vector, the Top-Level (Pseudo Non-Maskable) interrupt, the Divide by Zero Trap vector and, optionally, the interrupt vector table for use with the on-chip peripherals and the external interrupt sources. Each vector is contained in two consecutive byte locations, the high order address held in the lower (even) byte, the low order address held in the upper (odd) byte, forming the address which is loaded into the Program Counter when selected by the interrupt vector provided by the interrupt source. This should point to the relevant Interrupt Service routine provided by the User for immediate response to the interrupt.

1.2.1.2 Data Space

The ST9291 addresses the 640 bytes of on-chip RAM memory from addresses FD80h to FFFFh in both Program and Data Space. On-chip general purpose Registers may be used as additional RAM memory for minimum chip count systems.

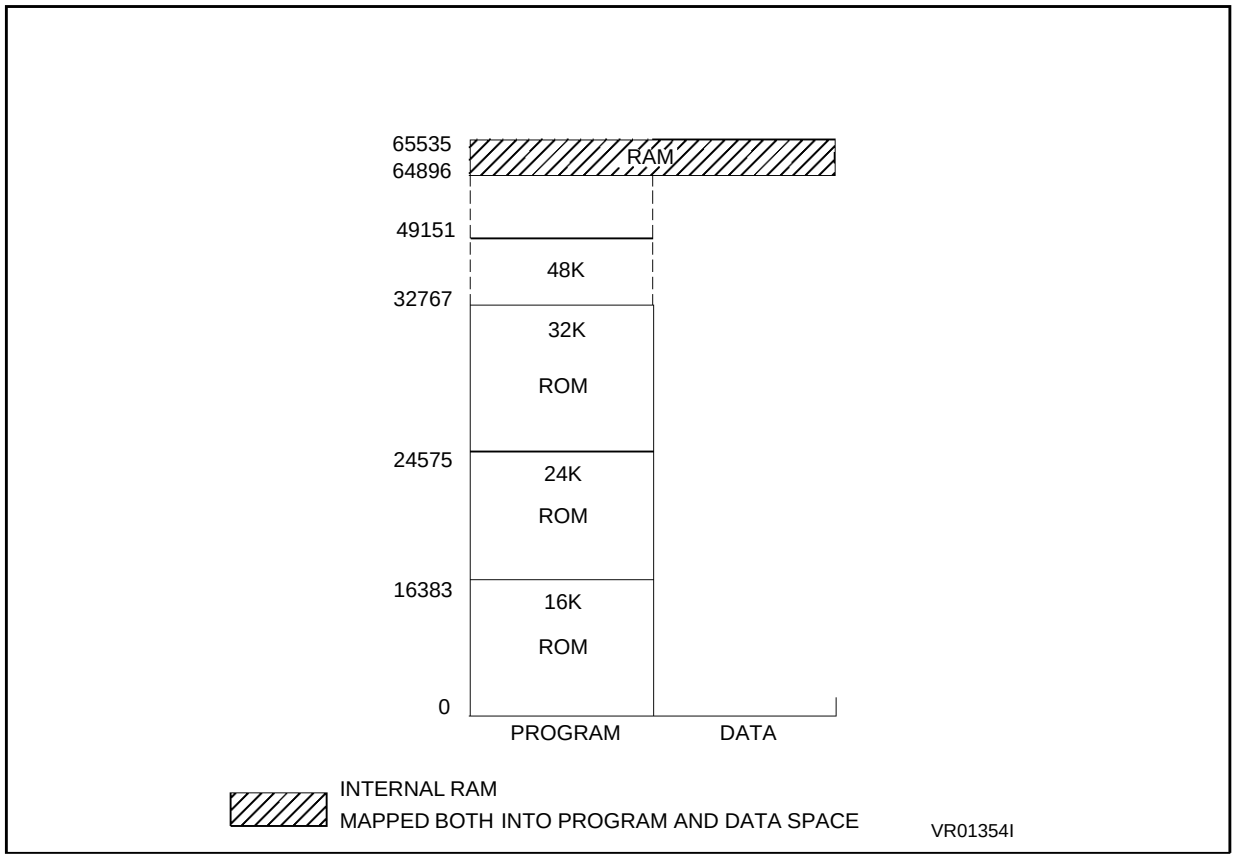
The Data Space is selected by the execution of the SDM instruction. All subsequent memory references will access the Data Space. When a separate Data Space is not required, data may stored in RAM or ROM memory within the Program Space.

MEMORY (Continued)

Table 1-4. ROM and RAM Address Configuration

Device Suffix	ROM Size (Bytes)	ROM Addresses		RAM Size (Bytes)	RAM Addresses	
J2/N2	16K	0 - 16383	dec	384	64896 - 65279	dec
		0000 - 3FFF	hex		FD80 - FEFF	hex
J3/N3	16K	0 - 16383	dec	640	64896 - 65535	dec
		0000 - 3FFF	hex		FD80 - FFFF	hex
J4/N4	24K	0 - 24575	dec	384	64896 - 65299	dec
		0000 - 5FFF	hex		FD80 - FEFF	hex
J5/N5	24K	0 - 24575	dec	640	64896 - 65535	dec
		0000 - 5FFF	hex		FD80 - FFFF	hex
J6/N6	32K	0 - 32767	dec	640	64896 - 65535	dec
		0000 - 7FFF	hex		FD80 - FFFF	hex
J7/N7	48K	00000 - 49151	dec	640	64896 - 65555	dec
		0000 - BFFF	hex		FD80 - FFFF	hex

Figure 1-13. ST9291 Memory Map



Notes: