SGS-Thomson Introduces 8-Bit µC

Full-Featured ST9 Family Challenges U.S., Japan

By Mark Thorson

SGS-Thomson's ST9 microcontroller family—previously available only to large European customers—is now being made available on the merchant semiconductor market in the U.S. While existing 8-bit architectures have the advantage of years of momentum, the ST9 has the benefit of a more powerful, modern (though still CISC) instruction set architecture.

Table 1 shows the current members of the ST9 family. In addition to any on-chip RAM shown in the table, each ST9 device contains a 224-byte general-purpose register file. All ST9 devices include a DMA controller, watchdog timer, and clocked serial interface as standard features.

The core CPU of the ST9 is a static implementation with a minimum instruction time (e.g., register-register add) of six ticks of the internal clock, i.e., 500 ns at 12 MHz (24 MHz oscillator clock divided by two). At the maximum clock rate, all ST9 parts are specified to draw 70 mA maximum, 32 mA typical. A mode entered by executing the halt instruction drops the power to 100 µA maximum.

Instruction Set

The instruction set includes a complete set of 8-bit arithmetic and logical operations, with 16-bit versions available for nearly all operations. Table 2 shows the instruction set. The processor is big-endian (i.e., most-significant byte at the lowest address).

Unsigned 8 × 8 multiply to a 16-bit product takes 1.83 µs at 24 MHz. Unsigned 16/8 divide to an 8-bit quotient and an 8-bit remainder takes 2.33 µs. An unsigned 32/16 divide-step instruction is available for producing a 16-bit quotient and 16-bit remainder. Division by zero causes a trap, which is unusual among 8-bit processors; most of them don't have any CPU exceptions for program errors. It's too bad the designers of the ST9 didn't go further and include traps for illegal instruction encoding and stack overflow/underflow, because of their value as a run-time "sanity check" for embedded code in noisy environments.

Boolean bit-manipulation operations are available between registers of the current working register set (see following description of the register organization). Both the register and the bit are directly specified by

Part Number	RAM	ROM	EPROM	EEPROM	A/D	USARTs	Timers	Parallel I/O	Package	Price
ST9026B6	256	16K	-	-	-	1	1	40	PDIP(48)	$6.90
ST90T26B6	256	-	16K	-	-	1	1	40	PDIP(48)	$11.00
ST90E26D6	256	-	16K	-	-	1	1	40	CDIP(48)	$25.00
ST9027B6	256	16K	-	-	-	1	1	32	PDIP(40)	$5.80
ST90T27B6	256	-	16K	-	-	1	1	32	PDIP(40)	$9.20
ST90E27D6	256	-	16K	-	-	1	1	32	CDIP(40)	$22.00
ST9028C6	256	16K	-	-	-	1	1	36	PLCC(44)	$6.20
ST90T28C6	256	-	16K	-	-	1	1	36	PLCC(44)	$9.90
ST90E28L6	256	-	16K	-	-	1	1	36	CLCC(44)	$60.00
ST9030C6	-	8K	-	-	1	1	2	56	PLCC(68)	$7.50
ST9030Q6	-	8K	-	-	1	1	2	56	PQFP(80)	$7.40
ST90T30C6	-	-	8K	-	1	1	2	56	PLCC(68)	$11.70
ST90T30Q6	-	-	8K	-	1	1	2	56	PQFP(80)	$11.60
ST90E30L6	-	-	8K	-	1	1	2	56	CLCC(68)	$46.00
ST90R30C6	-	-	-	-	1	1	2	40	PLCC(68)	$6.60
ST9040C6	256	16K	-	512	1	1	2	56	PLCC(68)	$13.70
ST9040Q6	256	16K	-	512	1	1	2	56	PQFP(80)	$13.40
ST90T40C6	256	-	16K	512	1	1	2	56	PLCC(68)	$22.60
ST90T40Q6	256	-	16K	512	1	1	2	56	PQFP(80)	$22.30
ST90E40L6	256	-	16K	512	1	1	2	56	CLCC(68)	$54.00
ST90R40C6	256	-	-	512	1	1	2	40	PLCC(68)	$11.90
ST90R50C6	-	-	-	-	1	2	3	56	PLCC(84)	$7.00

Table 1. ST9 family devices. All prices are 10K-unit quantity except windowed ceramic packages (90Exx parts) which are single-unit.

Byte Mnemonic	Word Mnemonic	Description
LD	LDW	Load
ADC	ADCW	Add with carry
ADD	ADDW	Add
SUB	SUBW	Subtract
SBC	SBCW	Subtract with carry
AND	ANDW	Logical AND
OR	ORW	Logical OR
CP	CPW	Compare
TM	TMW	Test under mask
TCM	TCMW	Test complement under mask
INC	INCW	Increment
DEC	DECW	Decrement
SLA	SLAW	Shift left arithmetic
SRA	SRAW	Shift right arithmetic
RRC	RRCW	Rotate right through carry
RLC	RLCW	Rotate left through carry
ROR		Rotate right
ROL		Rotate left
CLR		Clear
CPL		Complement
SWAP		Swap nibbles
DA		Decimal adjust
PUSH	PUSHW	Push on system stack
	PEA	Push effective address on system stack
POP	POPW	Pop from system stack
PUSHU	PUSHUW	Push on user stack
	PEAU	Push effective address on user stack
POPU	POPUW	Pop from user stack
MUL		Multiply
DIV		Divide
	DIVWS	Divide step
BLD		Bit load
BAND		Bit AND
BOR		Bit OR
BXOR		Bit XOR
BSET		Bit set
BRES		Bit reset
BCPL		Bit complement
BTSET		Bit test and set
CPJFI		Compare and jump if false else increment
CPJTI		Compare and jump if true else increment
BTJF		Bit test and jump if false
BTJT		Bit test and jump if true
DJNZ	DWJNZ	Decrement and jump if not zero
	JRcc	Conditional relative jump
	JPcc	Conditional absolute jump
	JP	Unconditional absolute jump
	CALL	Unconditional absolute subroutine call
	RET	Subroutine return
	IRET	Interrupt handler return
WFI		Wait for interrupt
HALT		Halt
XCH		Exchange
SRP		Set register pointer long
SRP0		Set register pointer 0
SRP1		Set register pointer 1
SPP		Set page pointer
EXT		Sign extend
EI		Enable interrupts
DI		Disable interrupts
SCF		Set carry flag
RCF		Reset carry flag
CCF		Complement carry flag
SPM		Select program memory
SDM		Select data memory
NOP		No operation

Table 2. ST9 instruction set.

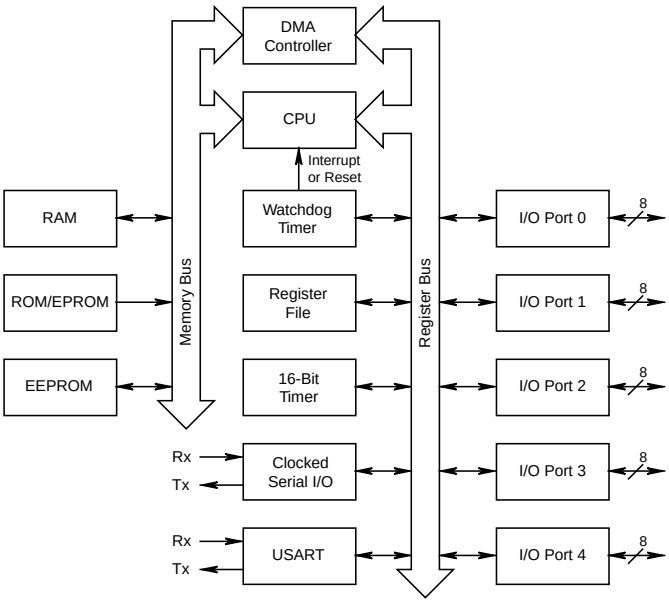

Figure 1. Block diagram of an ST9 microcontroller.

fields in the instruction format. A variant of the bit test instruction also allows specifying the bit position and register using a pointer in memory. Conditional branch instructions are available for testing the state of a working register bit. The registers for any of the on-chip peripherals can be selected as the working register set, which makes it easy to test status bits and change the state of control bits.

Several integrated loop control instructions are provided: decrement byte and jump if not zero, decrement word and jump if not zero, compare register against register-indirect memory location and jump if equal else increment register pointer, and compare register against register-indirect memory location and jump if not equal else increment register pointer. The latter instruction can be used to implement a one-instruction character-search loop.

Registers

The ST9 uses a register banking scheme similar to Zilog's Z8. Easy access is provided to a 16-register "working register set," slower access to the 256-byte on-chip register file, and slowest access to memory. It should be noted, however, that the ST9 provides easier access to memory than most other 8-bit microcontrollers because it allows a rich variety of addressing modes for the instruction operand fields.

The registers are organized as 16 banks of 16 registers. A dedicated pair of pointer registers selects one bank of 16 registers or two independent half-banks of eight registers as the working register set. Double-operand arithmetic and logical instructions are available in

both a two-byte format with 4-bit opcode fields for referencing the working registers, and a three-byte format with 8-bit fields which can directly address the entire register space. There are also formats that operate on a working register and a directly-addressed register, with either as the source or destination. When operating on register operands, a typical double-operand instruction in the two-byte format executes in six clock cycles, as compared to 10 cycles for the three-byte format.

Register banks 0 through C (hex) are completely general-purpose. Access to bank D (hex) is limited, because it is the escape code for specifying the working register set when the 8-bit direct-address operand format is used. Bank D cannot be referenced using direct addressing unless it is selected as the working register set. Together with bank D, there are a total of 224 byte-length general-purpose registers.

Bank E consists of dedicated CPU and system control registers. Bank F is paged among device registers for the on-chip peripherals; the page pointer register (a dedicated system control register) selects which peripheral appears in bank F.

There are two 16-bit stack pointers, system and user. The distinction is that the system stack is used to hold the return addresses for subroutine calls, interrupts, and exceptions, in addition to whatever use the user may make of it. The user stack lacks these automatic references, so it is more convenient for some uses of the stack such as passing parameters to a subroutine. There are no system/user privilege modes or protection levels. Software is responsible for avoiding stack overflow or underflow (there is no hardware stack limit checking). The CPU mode register has two bits for independently specifying the location of either stack in the register file or in memory. Stacks grow down, using predecrement/postincrement addressing.

Memory

There are separate 64K address spaces for program and data. Instructions are fetched from program space. Instruction operands are accessed in program space or data space depending on the state of the Data/Program bit in the CPU flags register. (The SDM and SPM instructions set and clear the bit, respectively.) Stacks are accessed in data space, if they are located in memory. Variants of the load instruction allow moving data directly between program and data space. On-chip ROM or EPROM is decoded as program space, and on-chip RAM and EEPROM (if any) is decoded as data space.

Off-chip memory and peripherals can be accessed by enabling the alternate functions for I/O ports. Two 8-bit ports can be enabled for a multiplexed bus with 8-bit data and 16-bit addressing, and some ST9 devices allow enabling three ports for non-multiplexed 8-bit data and16-bit address buses.

Price & Availability

The prices for ST9 parts are shown in Table 1. All parts are in volume production. The NRE charge on ROM-based parts is $3000; minimum order is 3K units. SGS-Thomson Microelectronics, 1000 E. Bell Rd., Phoenix, AZ 85022. Phone: 602/867-6100; fax: 602/867-6290.

A few ST9 parts have an additional port that can be enabled to expand the program and data spaces to 8 Mbytes each using a bank switching mechanism. When this mode is enabled, the low 32K of each address space is a common area (i.e., not bank switched), and the upper 32K can be remapped anywhere within an 8M space. There are four bank switch registers, one each for mapping CPU references to program space, CPU references to data space, DMA references to program space, and DMA references to data space.

Addressing Modes

Although most double-operand instruction formats require one operand to be a working register (4-bit operand field) or a directly addressed register (8-bit field), within this restriction a rich set of addressing modes is available for the other operand:

- *Immediate*—8-bit constant encoded in the instruction.
- *Register*—working register or directly addressed register. (Separate instruction formats for each.)
- *Register indirect*—register file location addressed by the contents of a working register.
- *Memory indirect*—memory location addressed by the contents of a pair of working registers. The address registers may optionally be predecremented or postincremented.
- *Memory indexed (8- or 16-bit offset)*—memory location addressed by the sum of an 8- or 16-bit immediate and the contents of a pair of working registers.

In addition, there are two addressing modes available only when the other operand is a working register:

- *Memory direct*—memory location addressed by a 16-bit immediate.
- *Memory indexed (register offset)*—memory location addressed by the sum of two pairs of working registers.

There are also two memory-memory instruction formats, in which both operands are referenced with memory-indirect addressing. The source is addressed with a working register pair, while the destination can be addressed with either a working register pair or a directly addressed register pair.

Vendor and Architecture	Part Number	Clock Freq. (MHz)	Min. Instr. (ns)	RAM	mask ROM	OTP ROM	EE-PROM	I/O Lines	Price 10K Qty
SGS-Thomson ST9	ST9030	12	500	224	8K			56	$7.40
	ST90T30	12	500	224		8K		56	$11.60
	ST9040	12	500	480	16K		512	56	$13.40
	ST90T40	12	500	480		16K	512	56	$22.30
Motorola 68HC11	68HC11E9	8.4	952	512	12K		512	38	$8.00
	68HC11K4	8.4	952	768	24K		640	66	$13.00
TI TMS370	TMS370C356	20	1400	512	16K			55	$8.27
	TMS370C056	20	1400	512	16K		512	54	$9.32
NEC K-Series	µPD78214	12	333	512	16K			54	$7.20
	µPD78P214	12	333	512		16K		54	$15.20
	µPD78244	12	333	512	16K		512	54	$19.75
Hitachi H8/500	HD6433308	10	200	512	16K			66	$7.00
	HD6473308	10	200	512		16K		66	$12.00
Siemens 8051	SAB C502	18	667	512	16K			32	$6.00
	SAB C503	18	667	256	8K			32	$5.25

Table 3. High-end 8-bit microcontroller cost comparison. All have A/D, except Siemens' C502.

Peripherals

The ST9 is unusual among high-end 8-bit microcontrollers in having an on-chip DMA controller, although NEC's K-Series processors have a DMA-like capability in their Peripheral Management Unit (PMU).

The ST9's DMA controller works only with the on-chip peripherals, not with devices added to the external bus by the system designer. Each DMA-capable peripheral has its own set of DMA address, control, and count registers. The DMA controller transfers data between the peripheral and either the register file or memory. A DMA transfer takes 8 cycles (667 ns) when used with the register file, or 1.333 µs when used with memory. The DMA controller can access either program or data memory.

A "swap mode" allows the DMA controller to work efficiently with double buffers. It maintains two sets of address pointers and transfer counters. When the DMA controller reaches the end of one block, it automatically switches to the other block and issues an end-of-block interrupt to the CPU. While the DMA controller is busy with one block, the interrupt service routine can then perform whatever housekeeping is necessary to prepare the next block.

All ST9 devices include at least one USART (some have two), with speeds up to 1.5 Mbaud/sec (synchronous), 750 Kbaud (asynchronous, external clock), or 375 Kbaud (asynchronous, internal clock). A separate clocked serial interface supports chip-to-chip serial interface standards, such as Philips' I^2C interface.

Most ST9 devices include an 8-channel, 8-bit A/D converter with a conversion time of 11 µs.

Various family members have one, two, or three 16-bit general-purpose timers, each equipped with an 8-bit prescaler, two 16-bit capture registers, and two 16-bit compare registers. The timer can interrupt the CPU or make a request to the DMA controller. There is also a separate 16-bit watchdog timer with an 8-bit prescaler. The latter can be used as a simple general-purpose timer or a watchdog timer; expiration of the timer can interrupt the CPU or reset the system.

There are 5, 7, or 9 byte-wide I/O ports. Each port line has several programmable options. It can be an input, output, bidirectional, or it can take on its alternate function (such as an A/D input or an external bus signal). The logic levels can be TTL or CMOS. Outputs can be push-pull, open drain, or open drain with weak pull-up (to save the need for external resistors).

Development Tools

SGS-Thomson is supporting the ST9 with an in-circuit emulator. It consists of a "mainframe" unit with an RS-232 interface for a terminal or host PC. Four "drawers" are available to adapt the mainframe to the different device types of the ST9 family. A complete emulator system costs from $8380 to $8810, depending on which drawer is supplied. Drawers are also available separately.

Three programmers are available for handling EPROM-based ST9 parts. These come in three different versions for the different device types and packaging options, ranging from $474 to $773. Both the emulator and programmers are available for rental (contact vendor for pricing).

A macro assembler, linker, and simulator are available for PC, Sun 3, SPARC, and VAX/VMS environments. With the optional C compiler, the DOS version of this package costs $830.

Conclusion

As a relatively new architecture, the ST9 was not constrained to follow the architectural mistakes of the past. Indeed, programmers will enjoy the rich set of memory addressing modes available for arithmetic and logical operands, the almost-complete symmetry between 8- and 16-bit instructions, and the freedom from

accumulator-bound architectures.

The ST9 family is sole-sourced, which was once a rarity among 8-bit microcontrollers but is now more common. SGS-Thomson and Siemens have a technology exchange agreement, under which Siemens received flash memory technology and rights to the ST9, while SGS-Thomson received rights to the 80C166 family of 16-bit microcontrollers. After serious consideration, Siemens decided not to second-source the ST9 family because of its overlap with Siemens' 8051-architecture products at the low end and its 80C166 at the high end.

Table 3 compares price and features for a number of high-end 8-bit microcontrollers and two of the recent enhanced versions of the 8051 architecture from Siemens.

The ST9 offers a peripheral set very similar to Motorola's full-featured 68HC11, but with the enhancement of an on-chip DMA controller. It also includes some memory configurations with a small amount of on-chip EEPROM—a standard feature of U.S. high-end 8-bit microcontrollers such as the 68HC11 and TI's TMS370, but rarely available in Japanese families. The ROM sizes range up to 16K, matching TI's largest parts, but smaller than Motorola (24K), Hitachi (32K), or NEC (32K).

The Siemens parts are not directly comparable with the other devices listed in the table, because their largest ROM size (16K) is not available with an on-chip A/D converter. All of the other devices in the table, including Siemens' 8K ROM part, have on-chip A/D. The Siemens parts are not available with OTP ROM. Siemens plans to fill the need for user-programmable devices by introducing parts with flash memory (expected in late 1993).

The ST9 is very competitive on speed. Although not quite as fast on a per-instruction basis as NEC's K2 family and Hitachi's H8, its superior addressing modes should narrow the performance gap. NEC's family is an accumulator-bound Z80 derivative, and Hitachi's family does not allow memory operands for arithmetic and logical instructions. The ST9 allows memory operands and lacks the accumulator bottleneck. Both SGS-Thomson and the Japanese far outpace most U.S. microcontrollers, such as Motorola's 68HC11 and TI's TMS370 families, in terms of performance.

With regard to pricing, the ST9 is competitive with both Japanese and U.S. vendors. Japanese microcontrollers tend to be cheaper than U.S. microcontrollers, but this mostly reflects a difference in pricing for parts with on-chip EEPROM. There are few Japanese parts with EEPROM, while there are few high-end, 8-bit U.S. parts without it (especially in the larger ROM sizes). The ST9 approaches aggressive Japanese pricing in EEPROM-less microcontrollers, and comes close to U.S. pricing in microcontrollers with EEPROM.♦

Windows NT

Continued from page 8

censing Windows NT and completing the port as soon as possible may be the best chance companies such as HP have of broadening their workstation markets.

Of all the workstation vendors, Sun seems to be in the most difficult position with respect to NT. Lately, Sun has not kept up with SGI, HP, and DEC in performance and has been thriving largely on a superior application base. If that application base migrates to NT, Sun may be forced to provide NT on its workstations. Embracing NT, however, would ruin the prospects of the Solaris OS on the 486 and seriously weaken the SunSoft subsidiary. If Sun does *not* do a port of NT, the company that so loudly touted the advantages of open systems and standards may find itself in the embarrassing position of being the one of the only significant RISC system vendors not offering Windows NT on its systems.

In the final analysis, the RISC vendors are at the mercy of application developers. If developers decide not to make versions for RISC processors available, no RISC systems will be sold. Part of the promise of Windows NT is that the consistent OS and development environment for a range of processor architectures will make porting easy enough that developers of Intel-based NT applications will provide RISC versions, even though the RISC architectures won't initially have a large enough installed base to justify much development effort. This could dramatically level the playing field, allowing several processor architectures to participate in the general-purpose PC market, each seeking a particular niche.

In the short to medium term, Intel clearly has the best position: whatever happens, the demand for performance is likely to increase, and x86 processors will be able to run whatever succeeds: Windows 3.x, Windows NT, or OS/2 (or NeXTStep, Solaris, or Taligent's Pink, for that matter). In the near term, most NT machines are likely to use 486 or P5 processors, not MIPS, Alpha, or any other RISC. Ironically, while NT is giving RISC-based computers their biggest chance to broaden their market, it will also give a big boost to the x86 family by offering PC users a vastly improved operating system and pulling some workstation business into the x86 fold.

Despite Intel's advantage, NT is giving RISC a chance. Now it is up to the systems vendors to make good on RISC's promises. If NT and Cairo fulfill the promises Microsoft has made, RISC vendors may be able to take market share away from Intel by exploiting portable, standardized operating systems and the advantages offered by RISC architectures—if they are able to deliver on the promise of better price/performance at high-volume price points.♦