SINGLE-CHIP 8-BIT MICROCOMPUTERS

DESCRIPTION

The MAB8400 family of single-chip 8-bit microcomputers are fabricated in N-MOS.

The family consists of 4 devices:

- **MAB8400**: capacity of 128 RAM bytes ('Piggy-back version'),
- **MAB8410**: 1K ROM/64 RAM bytes,
- **MAB8420**: 2K ROM/64 RAM bytes, [x])
- **MAB8440**: 4K ROM/128 RAM bytes.

Each type has 20 quasi-bidirectional I/O port lines, one serial I/O line, one single-level vectored interrupt, and an 8-bit timer/event counter and an on-board clock oscillator and clock circuits.

This microcomputer family is designed to be an efficient controller as well as an arithmetic processor. The instruction set is based on that of the MAB8048. The microcomputers have extensive bit handling ability and facilities for both binary and BCD arithmetic.

FEATURES

- 8-bit CPU, ROM, RAM, I/O in a single 28-lead DIL package
- 1K, 2K or 4K ROM bytes
- 64, 64 or 128 RAM bytes
- 20 quasi-bidirectional I/O port lines
- Two test inputs: one of which can be used to detect zero voltage cross-over, the other is also the external interrupt input
- Single-level vectored interrupts: external, timer/event counter, serial I/O
- Serial I/O which can be used in bus systems with more than one master (serial I/O data via an existing port line and clock via a dedicated line)
- 8-bit programmable timer/event counter
- Internal oscillator, clock driver
- Over 80 instructions (based on MAB8048)
- All instructions 1 or 2 cycles
- Single 5 V supply (± 10%)
- Operating temperature range: 0 to + 70 °C (MAB8400 family), or –40 to + 85 °C (MAF8400 family)

[x]) MAB 8421: same as MAB 8420,
but Port 1: $V_{0L} \leq 1{,}0$ V at I_{0L} = max. 10 mA

PACKAGE OUTLINES

MAB8400B : 28-lead 'Piggy-back' package (with up to 28-lead EPROM on top).
MAB8400Q: 56-lead QUIL; plastic (VO-35).
MAB8410P /20P /40P : 28-lead DIL; plastic (SOT-117D).
MAB8410D/20D/40D: 28-lead DIL; ceramic (SOT-135).
MAB8410T/20T/40T: 28-lead flat pack; plastic (SO-28; SOT-136A).

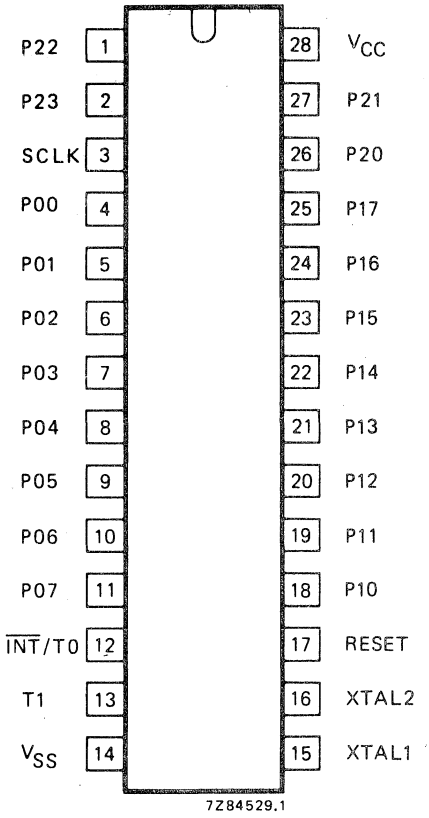


Fig. 1 Pinning diagram for MAB8400B 'Piggy-back' version bottom pinning (for top pinning see Fig. 2), MAB8410, MAB8420 and MAB8440.

PIN DESIGNATION

designation	pin no.	function
V_{SS}	14	**Ground**
V_{CC}	28	**Power supply**, + 5 V.
P00-P07	4-11	**Port 0**. 8-bit quasi-bidirectional I/O port.
P10-P17	18-25	**Port 1**. 8-bit quasi-bidirectional I/O port.
P20-P23	26,27,1,2	**Port 2**. 4-bit quasi-bidirectional I/O port; P23 is the serial data input/output in serial I/O mode.
SCLK	3	Bidirectional clock for serial I/O.
$\overline{INT}$/T0	12	External interrupt input (sensitive to negative-going edge); testable using the JT0, JNT0 instructions.
T1	13	Input pin testable using the JT1, JNT1 instructions. Can be designated the event counter input, using the STRT CNT instruction. Also allows zero cross-over sensing of slowly moving a.c. inputs.
RESET	17	Input, used to initialize the processor (active HIGH).
XTAL1	15	Connection to timing component (crystal) which determines the frequency of the internal oscillator. Also the input for an external clock source.
XTAL2	16	Connection of the other side of timing component.

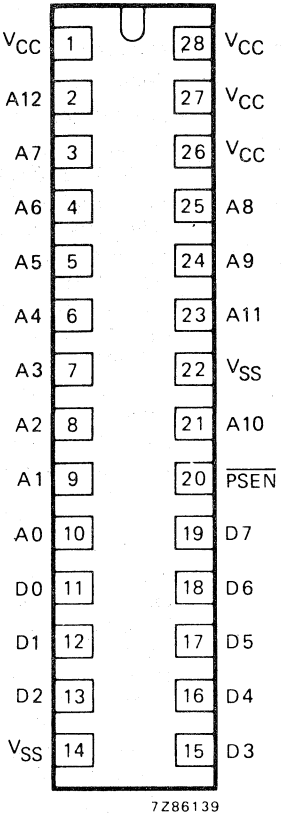


Fig. 2 Pinning diagram for MAB8400B 'Piggy-back' version top pinning (for bottom pinning see Fig. 1); to access a 2732 or 2764 EPROM.

PIN DESIGNATION

designation	pin no.	function
V_{SS}	14, 22	**Ground**
V_{CC}	1, 26-28	**Power supply, + 5 V**
A0-A12	10-3, 25, 24, 21, 23, 2	Address outputs
D0-D7	11-13, 15-19	Data
$\overline{PSEN}$	20	Program store enable

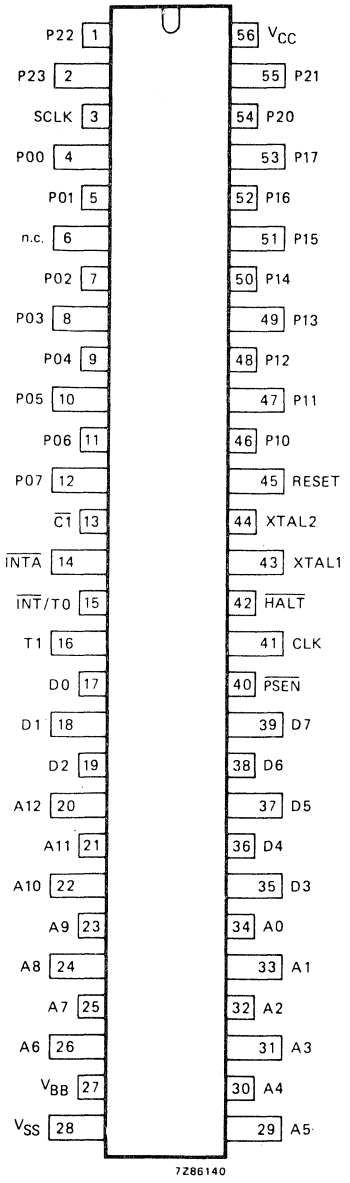


Fig. 3 Pinning diagram for MAB8400Q; 56-leads, quadruple in-line.

PIN DESIGNATION

designation	pin no.	function
V_{SS}	28	**Ground**
V_{CC}	56	**Power supply,** + 5 V
P00-P07	4, 5, 7-12	**Port 0.** 8-bit quasi-bidirectional I/O port.
P10-P17	46-53	**Port 1.** 8-bit quasi-bidirectional I/O port.
P20-P23	54, 55, 1, 2	**Port 2.** 4-bit quasi-bidirectional I/O port; P23 is the serial data input/output in serial I/O mode.
SCLK	3	Bidirectional clock for serial I/O.
$\overline{INT}$/T0	15	External interrupt input (sensitive to negative-going edge); testable using the JT0, JNT0 instructions.
T1	16	Input pin testable using the JT1, JNT1 instructions. Can be designated the event counter input, using the STRT CNT instruction. This also allows zero cross-over sensing of slowly moving a.c. inputs.
RESET	45	Input, used to initialize the processor (active HIGH).
XTAL1	43	Connection to timing component (crystal) which determines the frequency of the internal oscillator. Also the input for an external clock source.
XTAL2	44	Connection of the other side of timing component.
V_{BB}	27	Back bias voltage.

PIN DESIGNATION (continued)

designation	pin no.	function
A0-A12	34-29, 26-20	Program memory address outputs (active HIGH); A0 = LSB, A12 = MSB. Address output change after begin Phi3 of TS8.
D0-D7	17-19, 35-39	Data input lines (active HIGH). Used for reading external program memory. D0 = LSB, D7 = MSB.
CLK	41	Clock output buffered from XTAL2. On the positive-going edge the (internal) Phi clock goes HIGH.
$\overline{\text{PSEN}}$	40	Program store enable. This signal is used for enabling the external EPROM (e.g. on the 'Piggy-back' version). For emulation it enables the emulation memory and it indicates machine cycles. Active LOW during TS9, TS10 of each machine cycle and TS1 of the following machine cycle.
$\overline{\text{C1}}$	13	Cycle 1 indication output (active LOW). During emulation this signal indicates the opcode fetch cycle (useful for external instruction decoding, real time trace). Active from begin of TS10 of the cycle preceeding cycle 1, to begin TS10 of cycle 1.
$\overline{\text{HALT}}$	42	Halt input (active LOW). If activated, the current instruction is finished and the microcomputer stops execution (HALT mode). The next following program counter address is available on the address bus. Program counter and timer/counter are no longer updated. The serial I/O finishes the current transmit/receive action and goes into the idle state. Interrupts are not sampled in the HALT mode, they are only sampled when the microcomputer is running. Interrupt routines can be single-stepped as a normal program.
$\overline{\text{INTA}}$	14	Interrupt acknowledge output (active LOW). It indicates the acception of any interrupt. Active from begin of TS8 of the interrupted cycle, to begin of TS7 of the second cycle of the (internally forced) 'CALL vector address' instruction. During $\overline{\text{INTA}}$ active, the address bus shows the address, that has been saved in the stack (return address); the $\overline{\text{C1}}$ output indicates opcode fetch cycles as if a user CALL was executed.

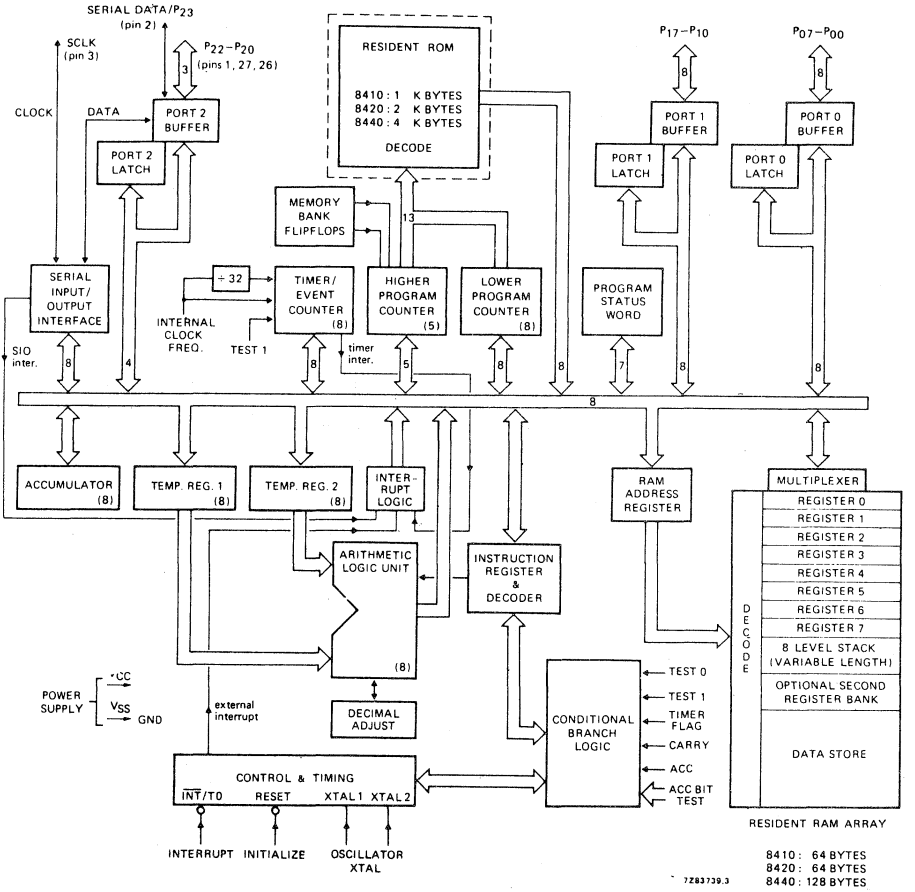


Fig. 4a Block diagram of the MAB8400 family.

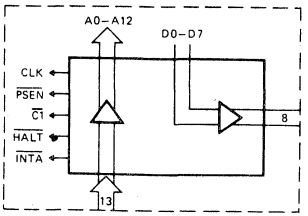


Fig. 4b Replacement of dotted part in Fig. 4a, showing the MAB8400Q bond-out version.

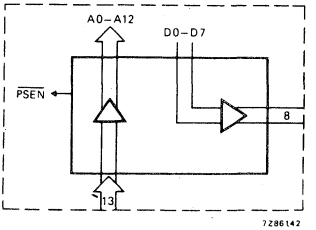


Fig. 4c Replacement of dotted part in Fig. 4a, for the MAB8400B 'Piggy-back' version.

FUNCTIONAL DESCRIPTION

Bond-out version MAB8400Q

The bond-out version is a microcomputer that contains no on-board ROM, but has all address and data lines brought-out to access an external ROM or EPROM. So, this version has more pins than the standard microcomputers with on-board ROM. It has all the features of the other members of the MAB8400 family. The RAM has 128 bytes.

Program memory (ROM)

The program memory consists of 1024, 2048, or 4096 bytes (8-bit words), which are addressed by the program counter. The memory is mask-programmed at our factory. Because the MAB8400 family offers a range of ROM capacities to suit the application, ROM expansion is not required. Figure 5 shows the program memory map.
Four program memory locations are of special importance:

location 0 – contains the first instruction to be executed after the processor is initialized (RESET),
location 3 – contains the first byte of an external interrupt service subroutine,
location 5 – contains the first byte of a serial I/O interrupt service subroutine,
location 7 – contains the first byte of a timer/event counter interrupt service subroutine.

Program memory is arranged in banks of 2K bytes, which are selected by SEL MB instructions. Further, the program memory is organized in pages of 256 bytes. Only the unconditional branch instructions (JMP and CALL) can cause jumps over page boundaries. Memory bank boundaries can be crossed only by using the same unconditional branch instructions after the appropriate memory bank has been selected. A CALL instruction can transfer control to a subroutine on any page; RET and RETR instructions can transfer control from a subroutine back to the main program. Note, a memory bank must be selected prior to a CALL or JMP instruction.

Data memory (RAM)

Data memory consists of 64 or 128 bytes (8-bit words). All locations are indirectly addressable using RAM pointer registers; up to 16 designated locations are directly addressable. Memory also includes an 8-level program counter stack addressed by a 3-bit stack pointer. Figure 6 shows the data memory map.

Locations 0 to 7 are designated as working registers, directly addressable by the direct register instructions. Because these registers are easily addressed and require the minimum instruction bytes to manipulate their contents, they are commonly used to store frequently accessed intermediate results. This bank of registers can be selected by the SEL RB0 instruction.

Executing the select register bank instruction SEL RB1, designates locations 24 to 31 as working registers, instead of locations 0 to 7, and these are then directly addressable. This second bank of working registers may be used as an extension of the first or reserved for use during interrupt service subroutines, saving the first bank for use in the main program. If the second bank is not used, locations 24 to 31 may serve as general purpose RAM.

The first two locations of each bank contain the RAM pointer registers R0, R1, R0' and R1', which indirectly address all RAM locations.

All RAM locations make efficient program loop counters when used with the decrement register and test instruction DJNZ.

Locations 8 to 23 may be designated as an 8-level program counter stack (2 locations per level), or as general purpose RAM. The program counter stack (Fig. 7) enables the processor to keep track of the return addresses and status generated by interrupts or CALL instructions by storing the contents of the program counter prior to servicing the subroutine. A 3-bit stack pointer determines which of the program counter stack's eight register pairs will be loaded with the next return address generated.

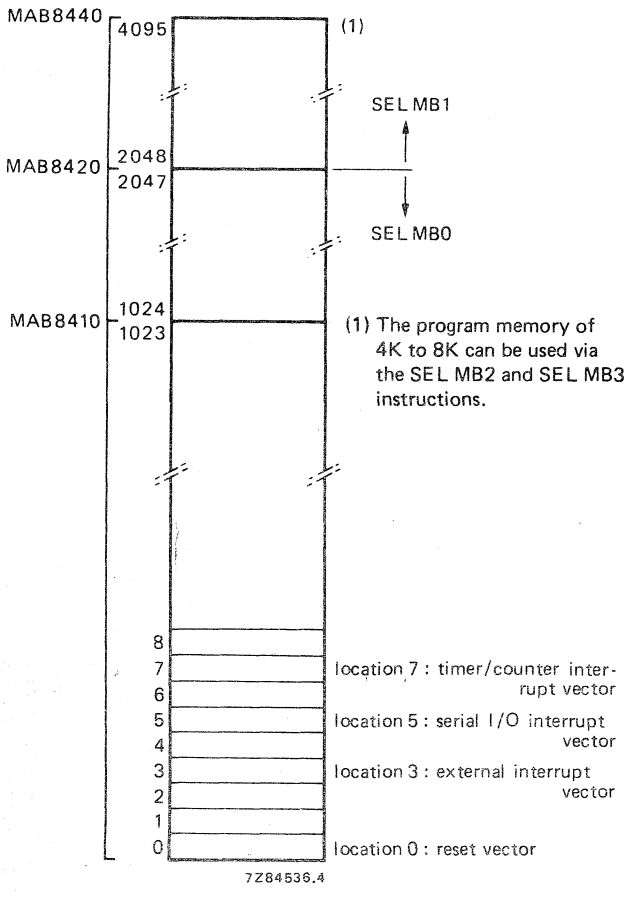


Fig. 5 Program memory map.

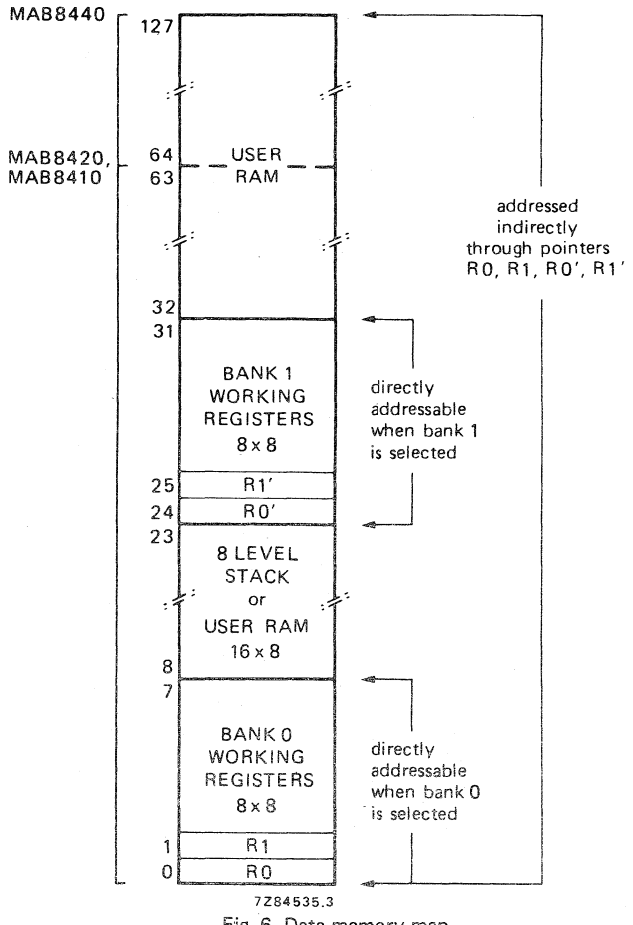


Fig. 6 Data memory map.

FUNCTIONAL DESCRIPTION (continued)

The stack pointer, when initialized to 000 by RESET, points to RAM locations 8 and 9. On the first subroutine CALL or interrupt, the contents of the program counter and bits 4, 6 and 7 of the program status word (PSW) are transferred to locations 8 and 9. The stack pointer increments by one and points to locations 10 and 11 ready for another call. Because an address may be up to 13 bits long, two bytes must be used to store each address.

At the end of a subroutine, which is signalled by a return instruction (RET or RETR), the stack pointer decrements by one and the contents of the register pair on top of the stack are transferred to the program counter. The saved PSW bits are transferred to the PSW only by the RETR instruction.

If not all 8 levels of subroutine and interrupt nesting are used, the unused portion of the stack may be used as any other indirectly addressable RAM locations. Possible locations from 32 to 127 may be used for storage of program variables or data.

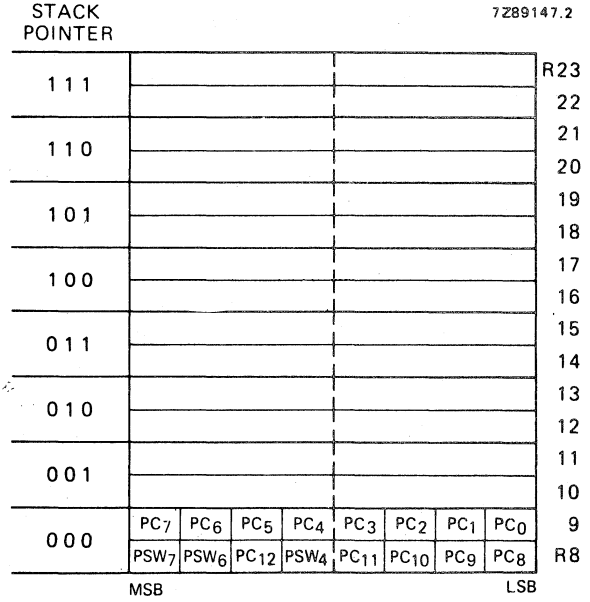


Fig. 7 Program counter stack.

Input/output

The MAB8400 family has 23 I/O lines arranged as:

- two parallel ports of 8 lines (P00 to P07, P10 to P17),
- one parallel port of 4 lines (P20 to P23),
- a serial I/O consisting of a data line shared with a parallel port line (P23) and a separate clock line SCLK,
- one external interrupt and test input ($\overline{INT}$/T0); when used as a test input it can be tested by the conditional branch instructions JT0 and JNT0,
- one test input (T1), which can alter program sequences when tested by conditional jump instructions JT1 and JNT1; T1 can also be used as an input to the timer/event counter, or for zero-cross detection.

FUNCTIONAL DESCRIPTION (continued)

Parallel ports

Output data written to a port is latched and remains unchanged until rewritten. Input data is not latched and so must be present until read by an input instruction.

Input lines are fully TTL compatible, output lines can drive one standard TTL load. Figure 8a shows the quasi-bidirectional I/O interface with push-pull output with pull-up resistance. Each line is continuously pulled up to + 5 V through a relatively high resistance ($\approx$ 50 kΩ). When a '0' is written to the line, the low impedance of TR1 overcomes the pull-up and provides TTL current sinking capability. When a '1' is written, TR2 is momentarily switched on to give fast pull-up. One state of a machine cycle later, the '1' level is maintained by the pull-up through the 50 kΩ. When used as an input line, a '1' must first be written to the line, otherwise the pull-down transistor TR1 is low impedance. This can be done by software control. After RESET, all I/O lines are in the input mode. The '1' on these lines can then be easily pulled down to a '0' by CMOS or TTL circuits.

Two other interface output configurations can also be specified:
- open drain output with pull-up (Fig. 8b),
- open drain output without pull-up (Fig. 8c),
- P23 only with open drain configuration.

Serial I/O

The MAB8400 family serial I/O interface has been designed to eliminate the heavy processing load imposed upon a normal microcomputer performing serial data transfer. Whereas a normal microcomputer must regularly monitor the serial data bus for the presence of data, the serial I/O interface detects, receives and converts the serial data stream into parallel format without interrupting the execution of the current program. An interrupt is sent to the microcomputer only when a complete byte is received. Then, the microcomputer reads the data byte in one instruction. Likewise, for transmission, the serial I/O interface performs parallel to serial conversion and subsequent serial output of the data and the microcomputer is only interrupted in the execution of its programmed tasks when a complete byte has been transmitted.

The design of the MAB8400 family serial I/O system allows any number of MAB8400 family devices to be interconnected by the two-line serial bus. The ability of any two devices to communicate, without interrupting the operation of any other devices on the bus, is an outstanding attribute of the system. This is achieved by allocating a specific 7-bit address to each device and providing a system whereby a device reacts only to message prefixed with its own address or the 'general call' address. Address recognition is performed by the interface hardware so that operation of the microcomputer need only be interrupted when a valid address has been received. This saves significant processing time and memory space compared with a convential microcomputer employing a software serial interface. When the addressing facility is not required, for instance in a system with only two microcomputers, direct data transfer without addressing can be performed. In multi-master systems, an automatically invoked arbitration procedure prevents two or more devices from continuing simultaneous transmission.

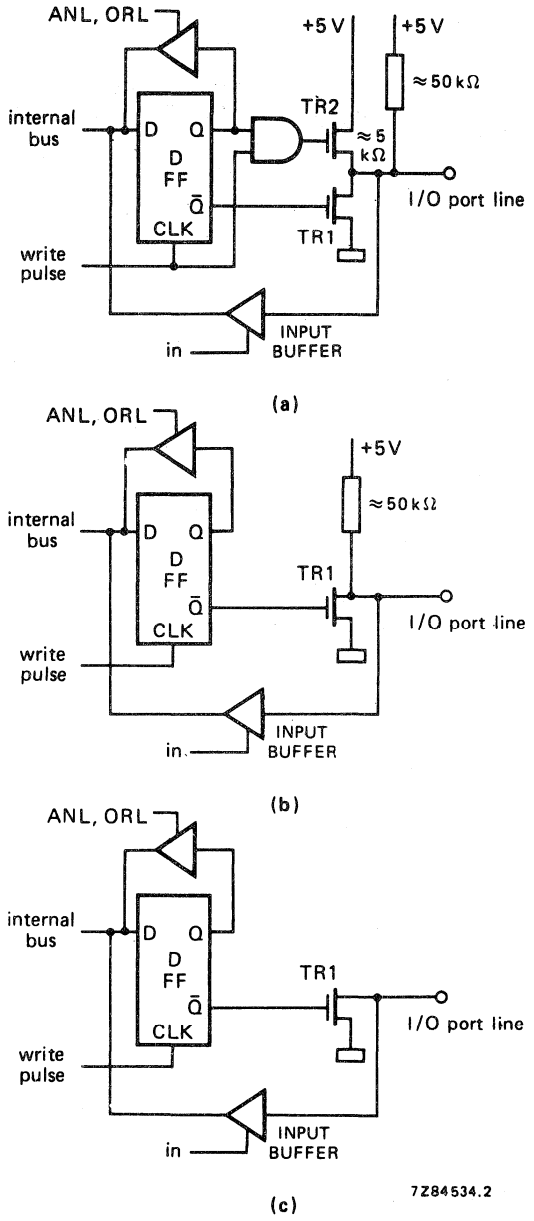


Fig. 8 Quasi-bidirectional I/O interface with (a) push-pull output with pull-up resistance; (b) open drain output with pull-up resistance; (c) open drain output without pull-up resistance.

FUNCTIONAL DESCRIPTION (continued)

Serial I/O interface

Figure 9 shows the serial I/O interface. The clock line of the serial bus has exclusive use of pin 3 (SCLK) while the data line shares pin 2 (serial data) with the I/O line P23 of port 2. When the serial I/O is enabled, P23 is disabled as a parallel port line; (P23 and SCLK only open drain).

The microcomputer and interface communicate via the internal microcomputer bus and the Serial Interrupt Request line. Data and information controlling the operation of the interface are stored in four registers:

- data shift register S0,
- serial I/O interface status word S1,
- serial clock control word S2,
- address register.

Data shift register.

S0 is the shift register that converts serial data to parallel format and vice versa. A pending interrupt is generated only after a complete byte has been transmitted, or after a complete data byte, specific address or general call address has been received. The most significant bit is transmitted first.

Status word S1.

S1 provides information about the state of the interface and stores interface control information from the microcomputer. Bits 0 to 3 are duplicated: control bits in these positions can only be written by the microcomputer, while interface status bits can only be read.

MST and TRX

These bits determine the operating mode of the serial I/O interface (Table 1).

Table 1 Operating modes of the serial I/O interface

MST	TRX	mode
0	0	slave receiver
1	0	master receiver
0	1	slave transmitter
1	1	master transmitter

BB: Bus Busy

This is the flag which indicates the status of the bus.

PIN: Pending Interrupt Not

PIN = '0' indicates the presence of a pending interrupt, which will cause a Serial Interrupt Request when the serial interrupt mechanism is enabled.

ESO: Enable Serial output

The ESO flag enables/disables the serial I/O interface: ESO = '1' enables, ESO = '0' disables.

BC0, BC1 and BC2

These bits indicate the number of bits received or transmitted in a serial data stream.

Bits ESO, BC0, BC1 and BC2 can only be written by software.

AL: Arbitration Lost

The arbitration lost flag is set by hardware when the serial I/O interface, as master transmitter, loses a bus arbitration procedure.

AAS: Addressed As Slave

This flag is set by hardware when the interface detects either its own specific address or the general call address as the first byte of a transfer and the interface has been programmed to operate in the • address recognition mode.

AD0: Address Zero

This flag is set by hardware after detection of the general call address when the interface is operating in the address recognition mode.

LRB: Last Received Bit

This contains either the last data bit received or, for a transmitting device in the acknowledgement mode, the acknowledgement signal from the receiving device.

Bits AL, AAS, AD0 and LRB can only be read by software.

Clock control register S2

Bits 0 to 4 of S2 are used to set the frequency of the serial clock signal. When a 4,43 MHz crystal is used, the frequency of the serial clock can be varied between 100 kHz and 720 Hz. An asymmetrical clock with a HIGH to LOW ratio of 3 : 1 can be generated using bit 5. The asymmetrical clock allows a microcomputer more time per clock period for sampling the data line, making the timing of this action less critical. Bit 6 can be used to activate the acknowledge mode of the serial I/O.
S2 is a write-only register.

Address register

The address register contains the 7-bit address back-up latches and the bit (ALS) used to enable/disable the address recognition mode. The address register can be written using the MOV S0, A and MOV S0, # data instructions, but only when ESO = '0'.

Serial I/O interrupt logic

An EN SI instruction enables and a DIS SI instruction disables the interrupt logic. When the logic is enabled, a pending interrupt results in a serial I/O interrupt to the processor, causing a call to location 5 in the ROM. When disabled, the presence of an interrupt is still indicated by PIN in S1, thus the interrupt can still be serviced. However, vectored interrupt will not occur.

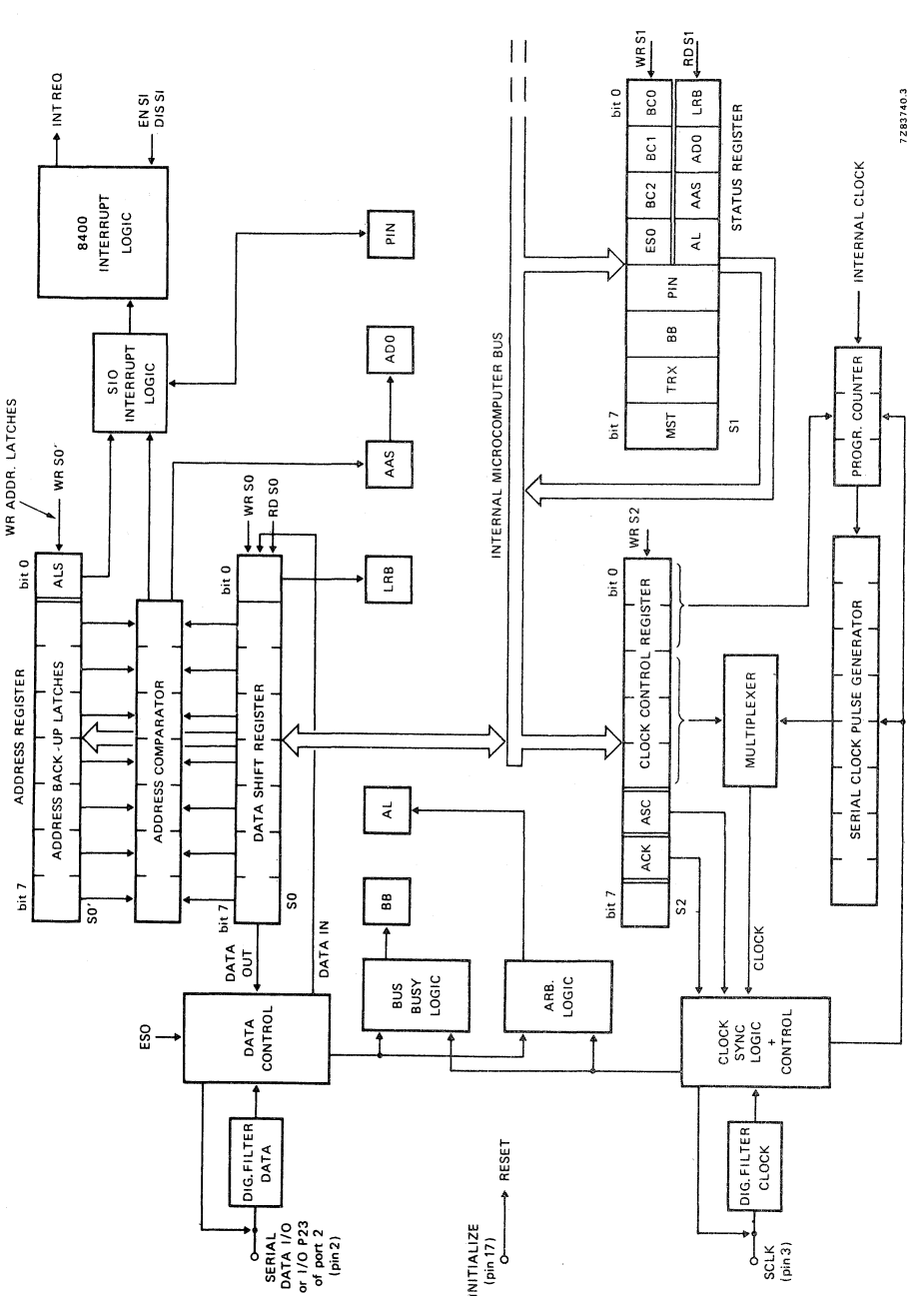


Fig. 9 Serial I/O interface.

FUNCTIONAL DESCRIPTION (continued)

Interrupt

When the external interrupt is enabled, a HIGH to LOW transition on the INT/T0 input initiates an external interrupt subroutine which causes a call to program memory location 3 following completion of the current instruction. The interrupt must remain enabled until the interrupt instruction is completed, otherwise the next instruction of the main program will be executed. Serial I/O interrupt, when enabled, causes a call to location 5, and a timer/event counter overflow a call to location 7, when the interrupt is enabled.

When the external interrupt is disabled, an external interrupt is latched. Therefore, keyboard or sensor interrupt requests are not lost when the processor must first perform some necessary functions whilst the external interrupt is disabled. When an interrupt subroutine starts, the program counter contents and bits 4, 6 and 7 of the PSW have been saved in the program counter stack. Accumulator contents have to be saved by software. Interrupt acknowledgement can be carried out by software via port pins. All interrupt routines must reside in memory bank 0.

The interrupt system is single-level – once an interrupt is detected, further interrupt requests are latched but ignored until the execution of a RETR instruction re-enables the interrupt input logic. After executing RETR, the program continues in the main part; this is independent of the occurrence of a second interrupt during the running of the first routine. If 2 or all 3 interrupts occur simultaneously, their priority is: (1) external, (2) serial I/O, (3) timer/event counter.

Another external interrupt can be created by enabling the timer/event counter interrupt loading FFH into the counter (one less than overflow) and enabling the event counter mode. A LOW to HIGH transition on the T1 input will then initiate an interrupt subroutine and cause a call to location 7.

Test input T1

The T1 input line can be used as:
- a test input for branch instructions,
- an input for zero voltage cross-over detection,
- an external input to the event counter.

A pull-up resistor can be provided as a ROM mask option. This is useful when the input is from a switch or a standard TTL output.

When T1 is used as a test input, the JT1 and JNT1 instructions test for 1 and 0 levels respectively. The T1 input has a self-biasing circuit which can detect when an a.c. signal crosses zero (within ± 100 mV when coupled through a 1 μF capacitor). The maximum input voltage is 3 V (peak-to-peak), the maximum frequency 1 kHz. Zero cross-over detection used in conjunction with the timer/event counter interrupt is useful in thyristor control of power equipment.

The operation of T1 as an input to the event counter is described under the heading *Timer/event counter.*

High current outputs

Four pins are provided which can sink higher currents (typical values):

– P23 (serial data, pin 2	5 mA at 0,45 V (open drain),
– SCLK, pin 3	5 mA at 0,45 V (open drain),
– P10, pin 18	7 mA at 2,5 V,
– P11, pin 19	7 mA at 2,5 V.

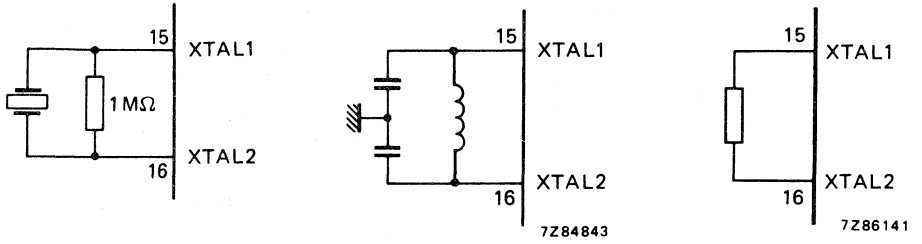
P10 and P11 may be paralled (if their logic outputs are always the same) to give 14 mA drive at 2,5 V.

FUNCTIONAL DESCRIPTION (continued)

Oscillator and clock

A crystal, inductor or resistor connected between XTAL1 and XTAL2 (see below) determines the frequency of the internal oscillator. An externally generated clock signal can also be applied to XTAL1 as the frequency reference. A machine cycle consists of 10 states, each state being 3 oscillator periods. The common 4,43 MHz crystal gives a 6,77 μs machine cycle.

The MAB8400 family has dynamic logic; for adequate refreshing the oscillator frequency must be $>$ 600 kHz.



Timer/event counter

An internal 8-bit binary up-counter is provided. This can count external events, modulo-32 machine cycles, or machine cycles directly (Fig. 10). Table 2 gives the instructions that control the counter and the prescaler and the functions performed.

When used as a timer, the input to the counter is either the overflow or input of a 5-bit prescaler. When used as an event counter, LOW to HIGH transitions on T1 (pin 13) are counted. The maximum rate at which the counter may be incremented is once every machine cycle (147,7 kHz for a 6,77 μs machine cycle). When the counter overflows, the timer flag is set. The flag can be tested and reset using the JTF (jump if timer flag = 1) instruction or JNTF instruction. Overflow also generates an interrupt to the processor when the timer/event counter interrupt is enabled.

Table 2 Timer/event counter control

function	timer mode modulo-1, module-32*	counter mode
CLEAR	MOV T,A (A) = 0 or RESET	MOV T,A (A) = 0 or RESET
PRESET	MOV T,A	MOV T,A
START	STRT T	STRT CNT
STOP	STOP TCNT or RESET	STOP TCNT or RESET
TEST	JTF/JNTF	JTF/JNTF
READ**	MOV A,T	MOV A,T

* With prescaler select, PS = 0, the timer counts modulo-32 machine cycles, with PS = 1 it counts modulo-1 cycles (prescaler not used); prescaler cleared with STRT T, prescaler not readable.

** READ does not disturb the counting process.

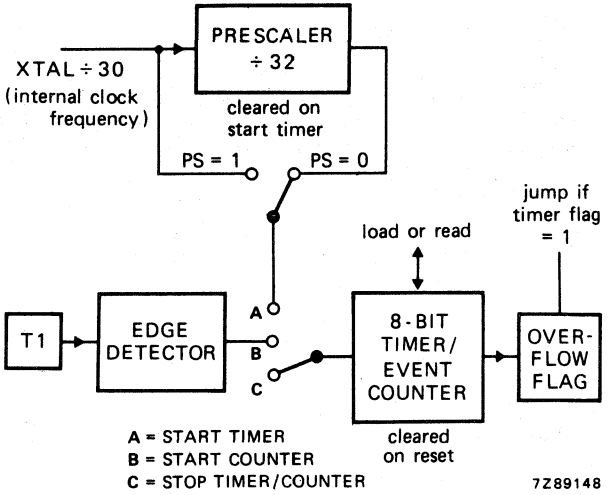


Fig. 10 Timer/event counter.

Program status word

The program status word (PSW) is an 8-bit word (1-byte) in the CPU which stores information about the current status of the microcomputer (Fig. 11). The PSW bits are:

bits 0, 1 and 2 – stack pointer bits (SP_0, SP_1, SP_2),
bit 3 – prescaler select (PS); 0 = modulo-32; 1 = modulo-1 (no prescaling),
bit 4 – working register bank select (RBS); 0 = register bank 0; 1 = register bank 1,
bit 5 – not used (1),
bit 6 – auxiliary carry (AC); half-carry bit generated by an ADD instruction and used by the decimal adjust instruction DA A,
bit 7 – carry (CY); the carry flag indicates that the previous operation has resulted in an overflow of the accumulator.

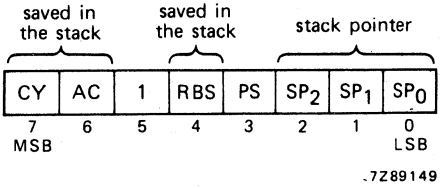


Fig. 11 Program status word.

All bits can be read using the MOV A, PSW instruction. Bits 7 and 6 are set and cleared by CPU operation. Bit 4 can be changed by a SEL RB instruction, bit 3 by the MOV PSW, A instruction, and bits 0, 1 and 2 by the CALL, RET or RETR instructions and in case of an interrupt. Bits 7, 6 and 4 are stored in the program counter stack during subroutine and interrupt calls. These bits are restored in the PSW with a RETR (return and restore) instruction which must be used at the end of an interrupt and can be used at the end of a normal subroutine. The RET instruction has no restore feature and cannot be used at the end of an interrupt.

FUNCTIONAL DESCRIPTION (continued)

Program counter

A 13-bit program counter is used so that up to 8K bytes of ROM can be addressed. Figure 12 shows the arrangement of the bits. During an interrupt subroutine PC_{11} and PC_{12} are forced to 0. All 13 bits are saved in the stack during CALL and interrupt routines.

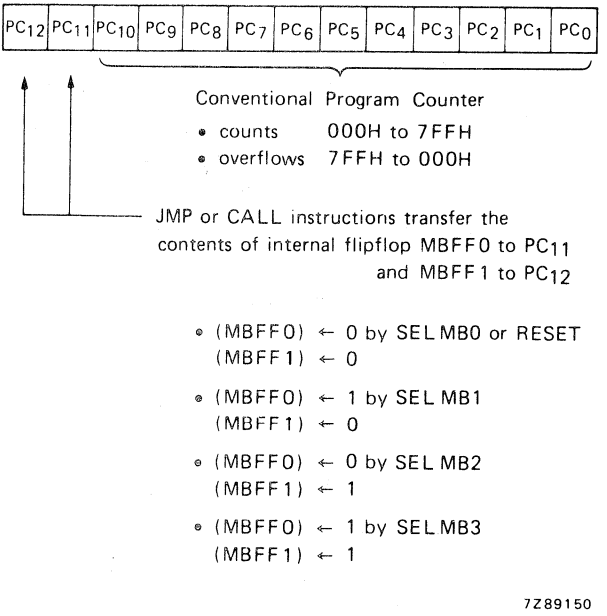


Fig. 12 Program counter.

Central processing unit

The MAB8400 family has arithmetic, logical and branching capabilities. The DA A, SWAP A, and XCHD instructions simplify BCD arithmetic and the handling of nibbles. The MOVP A,@A instruction permits efficient table look up from the current ROM page.

The conditional branch logic within the processor enables several conditions, internal and external to the processor, to be tested by the user's program. Table 3 lists the conditional jump instructions used to change the program execution sequence. The DJNZ instruction decrements a designated register or data memory location and branches if the contents are not zero. This instruction is useful for looping control. The JMPP@A instruction allows multiway branches to destinations determined by the contents of the accumulator.

Table 3 Conditional branches

test	jump condition	jump instruction
accumulator	0 or non-zero	JZ, JNZ
accumulator bit test	1	JB0 to JB7
carry flag	0 or 1	JNC, JC
timer overflow flag	0 or 1	JNTF, JTF
test input T0	0 or 1	JNT0, JT0
test input T1	0 or 1	JNT1, JT1
register	non-zero	DJNZ

Reset

A positive-going signal on the RESET input:
- sets the program counter to zero,
- selects location 0 of memory bank 0, and register bank 0,
- sets the stack pointer to zero (000); pointing to RAM address 8,
- disables the interrupts (external, timer and serial I/O),
- stops the timer/event counter, then sets it to zero,
- sets the timer prescaler to modulo-32,
- resets the timer flag,
- sets all ports to logic '1' (input mode),
- sets the serial I/O to slave receiver mode and disables the serial I/O.

The external power-on-reset circuit can consist of a capacitor connected between V_{CC} and the RESET pin. A diode may be added between the RESET pin and ground to endure reset if the supply voltage falls momentarily.
RESET has to be active HIGH for at least >2 machine cycles after the power supply and clock have stabilized.

TABLE 4. MAB8400 INSTRUCTION MAP

first hexadecimal character of opcode ↓

second hexadecimal character of opcode →

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP			ADD A,#data	JMP page 0	EN I	JNTF addr	DEC A	IN A,Pp				MOV A,Sn			
									0	1	2		0	1		
1	INC @Rr		JBO addr	ADDC A,#data	CALL page 0	DIS I	JTF addr	INC A	INC Rr							
	0	1							0	1	2	3	4	5	6	7
2	XCH A,@Rr			MOV A,#data	JMP page 1	EN TCNTI	JNTO addr	CLR A	XCH A,Rr							
	0	1							0	1	2	3	4	5	6	7
3	XCHD A,@Rr		JB1 addr		CALL page 1	DIS TCNTI	JTO addr	CPL A	OUTL Pp,A				MOV Sn,A			
	0	1							0	1	2		0	1	2	
4	ORL A,@Rr		MOV A,T	ORL A,#data	JMP page 2	STRT CNT	JNT1 addr	SWAP A	ORL A,Rr							
	0	1							0	1	2	3	4	5	6	7
5	ANL A,@Rr		JB2 addr	ANL A,#data	CALL page 2	STRT T	JT1 addr	DA A	ANL A,Rr							
	0	1							0	1	2	3	4	5	6	7
6	ADD A,@Rr		MOV T,A		JMP page 3	STOP TCNT		RRC A	ADD A,Rr							
	0	1							0	1	2	3	4	5	6	7
7	ADDC A,@Rr		JB3 addr		CALL page 3			RR A	ADDC A,Rr							
	0	1							0	1	2	3	4	5	6	7
8				RET	JMP page 4	EN SI			ORL Pp,#data							
									0	1	2					
9			JB4 addr	RETR	CALL page 4	DIS SI	JNZ addr	CLR C	ANL Pp,#data				MOV Sn,#data			
									0	1	2		0	1	2	
A	MOV @Rr,A			MOVP A,@A	JMP page 5	SEL MB2		CPL C	MOV Rr,A							
	0	1							0	1	2	3	4	5	6	7
B	MOV @Rr,#data		JB5 addr	JMPP @A	CALL page 5	SEL MB3			MOV Rr,#data							
	0	1							0	1	2	3	4	5	6	7
C	DEC @Rr				JMP page 6	SEL RB0	JZ addr	MOV A,PSW	DEC Rr							
	0	1							0	1	2	3	4	5	6	7
D	XRL A,@Rr		JB6 addr	XRL A,#data	CALL page 6	SEL RB1		MOV PSW,A	XRL A,Rr							
	0	1							0	1	2	3	4	5	6	7
E	DJNZ @Rr,addr				JMP page 7	SEL MB0	JNC addr	RL A	DJNZ Rr, addr							
	0	1							0	1	2	3	4	5	6	7
F	MOV A,@Rr		JB7 addr		CALL page 7	SEL MB1	JC addr	RLC A	MOV A,Rr							
	0	1							0	1	2	3	4	5	6	7

Instruction set

The MAB8400 family instruction set consists of over 80 one and two byte instructions and is based on the MAB8048 instruction set. New instructions include those for serial I/O operation and memory bank selection. Program code efficiency is high because all RAM locations and all ROM locations on a 256 byte page require only a single byte address.

Table 7 gives the instruction set of the MAB8400 family; Table 4 shows the instruction map. The following symbols and abbreviations are used:

symbol	description
A	accumulator
addr	program memory address
Bb	bit designation (b = 0–7)
RBS	register bank select
C	carry (bit CY)
CNT	event counter
D	mnemonic for 4-bit digit (nibble)
data	8-bit number or expression
I	interrupt
MB	memory bank
MBFF	memory bank flip-flop
P	mnemonic for 'in-page' operation
PC	program counter
Pp	port designation (p = 0, 1, 2)
PSW	program status word
RB	register bank
Rr	register designation (r = 0–7)
S_n	serial I/O register
SP	stack pointer
T	timer
TF	timer flag
T1	test 1 input
T0	test 0 input
#	immediate data prefix
@	indirect address prefix
(X)	contents of X
((X))	contents of location addressed by X
←	is replaced by
↔	is exchanged with

FUNCTIONAL DESCRIPTION (continued)

Table 5 shows the MAB8400 family instructions (including the five instructions for serial I/O operation) absent from the MAB8048 instruction set.

Table 5 MAB8400 family instructions not in the MAB8048 instruction set

serial I/O	register	control	conditional branch

MOV A, S_n
MOV S_n, A
MOV S_n, #data
EN SI
DIS SI
	DEC @Rr
	DJNZ @Rr, addr
		SEL MB2
		SEL MB3
			JNTF addr

Table 6 shows the MAB8048 instructions absent from the MAB8400 family instruction set.

Table 6 MAB8048 instructions not in the MAB8400 family instruction set

data moves	flags	branch	control

MOVX A, @R
MOVX @R, A
MOVP3 A, @A
MOVD A, P
MOVD P, A
ANLD P, A
ORLD P, A
	CLR F0
	CPL F0
	CLR F1
	CPL F1
		* JNI addr
		JF0 addr
		JF1 addr
		
		* replaced by JT0, JNT0.
			ENT0 CLK

Differences between the MAB8021, MAB8048 microcomputers and the MAB8400 family:

	8021	8048	8410, 8420, 8440, 8400
ROM capacity (bytes)	1K	1K	1K, 2K, 4K, ROMless
RAM capacity (bytes)	64	64	64, 64, 128, 128
parallel I/O lines	8 + 8 + 4	8 + 8 + 8	8 + 8 + 4
single inputs	1	3	2
serial I/O	no	no	yes, 2-line multi-transmitter
timer	8 bit	8 bit	8 bit
prescaler	mod. 32	mod. 32	mod. 1 & mod. 32
machine cycle time (μs)	10	2,5	6,7
for clock (MHz)	3	6	4,43
instruction set	8021	8048	8048 with omissions; 5 new serial I/O instructions; 2 new register instructions; 2 new control instructions; 1 new cond. branch instruction
interrupts	none	2 external timer/ event counter	3 external serial I/O timer/event counter
no. of pins (DIL)	28	40	28

Table 7 Instruction set is shown on the next 5 pages.

	mnemonic	opcode (hex.)	bytes/ cycles	description	function		notes
ACCUMULATOR	ADD A, Rr	6*	1/1	Add register contents to A	(A)←(A) + (Rr)	r = 0–7	1
	ADD A, @Rr	60 61	1/1	Add RAM data, addressed by Rr, to A	(A)←(A) + ((R0)) (A)←(A) + ((R1))		1
	ADD A, #data	03 data	2/2	Add immediate data to A	(A)←(A) + data		1
	ADDC A, Rr	7*	1/1	Add carry and register contents to A	(A)←(A) + (Rr) + (C)	r = 0–7	1
	ADDC A, @Rr	70 71	1/1	Add carry and RAM data, addressed by Rr, to A	(A)←(A) + ((R0)) + (C) (A)←(A) + ((R1)) + (C)		1
	ADDC A, #data	13 data	2/2	Add carry and immediate data to A	(A)←(A) + data + (C)		1
	ANL A, Rr	5*	1/1	'AND' Rr with A	(A)←(A) AND (Rr)	r = 0–7	
	ANL A, @Rr	50 51	1/1	'AND' RAM data, addressed by Rr, with A	(A)←(A) AND ((R0)) (A)←(A) AND ((R1))		
	ANL A, #data	53 data	2/2	'AND' immediate data with A	(A)←(A) AND data		
	ORL A, Rr	4*	1/1	'OR' Rr with A	(A)←(A) OR (Rr)	r = 0–7	
	ORL A, @Rr	40 41	1/1	'OR' RAM data, addressed by Rr, with A	(A)←(A) OR ((R0)) (A)←(A) OR ((R1))		
	ORL A, #data	43 data	2/2	'OR' immediate data with A	(A)←(A) OR data		
	XRL A, Rr	D*	1/1	'XOR' Rr with A	(A)←(A) XOR (Rr)	r = 0–7	
	XRL A, @Rr	D0 D1	1/1	'XOR' RAM, addressed by Rr, with A	(A)←(A) XOR ((R0)) (A)←(A) XOR ((R1))		
	XRL A, #data	D3 data	2/2	'XOR' immediate data with A	(A)←(A) XOR data		
	INC A	17	1/1	increment A by 1	(A)←(A) + 1		
	DEC A	07	1/1	decrement A by 1	(A)←(A) −1		
	CLR A	27	1/1	clear A to zero	(A)←0		
	CPL A	37	1/1	one's complement A	(A)←NOT(A)		
	RL A	E7	1/1	rotate A left	$(A_{n+1})\leftarrow(A_n)$ $(A_0)\leftarrow(A_7)$	n = 0–6	

	mnemonic	opcode (hex.)	bytes/ cycles	description	function		notes
ACCUMULATOR (cont.)	RLC A	F7	1/1	rotate A left through carry	$(A_{n+1})\leftarrow A_n$ $(A_0)\leftarrow(C)$, $(C)\leftarrow(A_7)$	n = 0–6	2
	RR A	77	1/1	rotate A right	$(A_n)\leftarrow(A_{n+1})$ $(A_7)\leftarrow(A_0)$	n = 0–6	
	RRC A	67	1/1	rotate A right through carry	$(A_n)\leftarrow(A_{n+1})$ $(A_7)\leftarrow(C)$, $(C)\leftarrow(A_0)$	n = 0–6	2
	DA A	57	1/1	decimal adjust A			2
	SWAP A	47	1/1	swap nibbles of A	$(A_{4-7})\leftrightarrow(A_{0-3})$		
DATA MOVES	MOV A, Rr	F*	1/1	move register contents to A	(A)←(Rr)	r = 0–7	
	MOV A, @Rr	F0 F1	1/1	move RAM data, addressed by Rr, to A	(A)←((R0)) (A)←((R1))		
	MOV A, #data	23 data	2/2	move immediate data to A	(A)←data		
	MOV Rr, A	A*	1/1	move accumulator contents to register	(Rr)←(A)	r = 0–7	
	MOV @Rr, A	A0 A1	1/1	move accumulator contents to RAM location addressed by Rr	((R0))←(A) ((R1))←(A)		
	MOV Rr, #data	B* data	2/2	move immediate data to Rr	(Rr)←data		
	MOV @Rr, #data	B0 data B1 data	2/2	move immediate data to RAM location addressed by Rr	((R0))←data ((R1))←data		
	XCH A, Rr	2*	1/1	exchange accumulator contents with Rr	(A)↔(Rr)	r = 0–7	
	XCH A, @Rr	20 21	1/1	exchange accumulator contents with RAM data addressed by Rr	(A)↔((R0)) (A)↔((R1))		
	XCHD A, @Rr	30 31	1/1	exchange lower nibbles of A and RAM data addressed by Rr	$(A_{0-3})\leftrightarrow((R0_{0-3}))$ $(A_{0-3})\leftrightarrow((R1_{0-3}))$		
	MOV A, PSW	C7	1/1	move PSW contents to accumulator	(A)←(PSW)		
	MOV PSW, A	D7	1/1	move accumulator bit 3 to PSW_3	$(PSW_3)\leftarrow(A_3)$		3
	MOVP A, @A	A3	1/2	move indirectly addressed data in current page to A	$(PC_{0-7})\leftarrow(A)$, $(A)\leftarrow((PC))$		

FLAGS	CLR C	97	1/1	clear carry bit	(C)←0		2
	CPL C	A7	1/1	complement carry bit	(C)←NOT(C)		2
REGISTER	INC Rr	1*	1/1	increment register by 1	(Rr)←(Rr) + 1	r = 0–7	
	INC @Rr	10 11	1/1	increment RAM data, addressed by Rr, by 1	((R0))←((R0)) + 1 ((R1))←((R1)) + 1		
	DEC Rr	C*	1/1	decrement register by 1	(Rr)←(Rr) − 1	r = 0–7	
	DEC @Rr	C0 C1	1/1	decrement RAM data, addressed by Rr, by 1	((R0))←((R0)) − 1 ((R1))←((R1)) − 1		
BRANCH	JMP addr	●4 address	2/2	unconditional jump within a 2Kbank	(PC_{8-10})←$addr_{8-10}$ (PC_{0-7})←$addr_{0-7}$ (PC_{11-12})←MBFF 0–1		
	JMPP @A	B3	1/2	indirect jump within a page	(PC_{0-7})←((A))		
	DJNZ Rr, addr	E* address	2/2	decrement Rr by 1 and jump if not zero to addr	(Rr)←(Rr) − 1 if (Rr) not zero (PC_{0-7})←addr	r = 0–7	
	DJNZ @Rr, addr	E0	2/2	decrement RAM data, addressed by Rr, by 1 and jump if not zero to addr	((R0))←((R0)) − 1 if ((R0)) not zero (PC_{0-7})←addr		
		E1			((R1))←((R1)) − 1 if ((R1)) not zero (PC_{0-7})←addr		
	JBb addr	▲2 address	2/2	jump to addr if Acc. bit b = 1	if b = 1: (PC_{0-7})←addr	b = 0–7	
	JC addr	F6 address	2/2	jump to addr if C = 1	if C = 1: (PC_{0-7})←addr		
	JNC addr	E6 address	2/2	jump to addr if C = 0	if C = 0: (PC_{0-7})←addr		
	JZ addr	C6 address	2/2	jump to addr if A = 0	if A = 0: (PC_{0-7})←addr		
	JNZ addr	96 address	2/2	jump to addr if A is NOT zero	if A ≠ 0: (PC_{0-7})←addr		
	JT0 addr	36 address	2/2	jump to addr if T0 = 1	if T0 = 1: (PC_{0-7})←addr		
	JNT0 addr	26 address	2/2	jump to addr if T0 = 0	if T0 = 0: (PC_{0-7})←addr		
	JT1 addr	56 address	2/2	jump to addr if T1 = 1	if T1 = 1: (PC_{0-7})←addr		
	JNT1 addr	46 address	2/2	jump to addr if T1 = 0	if T1 = 0: (PC_{0-7})←addr		
	JTF addr	16 address	2/2	jump to addr if Timer Flag = 1	if TF = 1: (PC_{0-7})←addr		4
	JNTF addr	06 address	2/2	jump to addr if Timer Flag = 0	if TF = 0: (PC_{0-7})←addr		

	mnemonic	opcode (hex.)	bytes/ cycles	description	function	notes
TIMER/EVENT COUNTER	MOV A, T	42	1/1	move timer/event counter contents to accumulator	(A)←(T)	
	MOV T, A	62	1/1	move accumulator contents to timer/event counter	(T)←(A)	
	STRT CNT	45	1/1	start event counter		
	STRT T	55	1/1	start timer		
	STOP TCNT	65	1/1	stop timer/event counter		
	EN TCNTI	25	1/1	enable timer/event counter interrupt		
	DIS TCNTI	35	1/1	disable timer/event counter interrupt		
CONTROL	EN I	05	1/1	enable external interrupt		
	DIS I	15	1/1	disable external interrupt		
	SEL RB0	C5	1/1	select register bank 0	(RBS)←0	5
	SEL RB1	D5	1/1	select register bank 1	(RBS)←1	5
	SEL MB0	E5	1/1	select program memory bank 0	(MBFF0)←0, (MBFF1)←0	
	SEL MB1	F5	1/1	select program memory bank 1	(MBFF0)←1, (MBFF1)←0	
	SEL MB2	A5	1/1	select program memory bank 2	(MBFF0)←0, (MBFF1)←1	
	SEL MB3	B5	1/1	select program memory bank 3	(MBFF0)←1, (MBFF1)←1	
SUBROUTINE	CALL addr	▲4 address	2/2	jump to subroutine	((SP))←(PC), ($PSW_{4, 6, 7}$) (SP)←(SP) + 1 (PC_{8-10})←$addr_{8-10}$ (PC_{0-7})←$addr_{0-7}$ (PC_{11-12})←MBFF 0–1	6
	RET	83	1/2	return from subroutine	(SP)←(SP) – 1 (PC)←((SP))	6
	RETR	93	1/2	return from interrupt and restore bits 4, 6, 7 of PSW	(SP)←(SP) – 1 ($PSW_{4, 6, 7}$) + (PC)←((SP))	6

PARALLEL INPUT/OUTPUT	IN A, Pp	08 09 0A	1/2	input port p data to accumulator	(A)←(P0) (A)←(P1) (A)←(P2)	7
	OUTL Pp, A	38 39 3A	1/2	output accumulator data to port p	(P0)←(A) (P1)←(A) (P2)←(A)	
	ANL Pp, #data	98 99 9A	2/2	AND port p data with immediate data	(P0)←(P0) AND data (P1)←(P1) AND data (P2)←(P2) AND data	
	ORL Pp, #data	88 89 8A	2/2	OR port p data with immediate data	(P0)←(P0) OR data (P1)←(P1) OR data (P2)←(P2) OR data	
SERIAL INPUT/OUTPUT	MOV A, S_n	0C 0D	1/2	move serial I/O register contents to accumulator	(A)←(S0) (A)←(S1)	8
	MOV S_n, A	3C 3D 3E	1/2	move accumulator contents to serial I/O register	(S0)←(A) (S1)←(A) (S2)←(A)	
	MOV S_n, #data	9C 9D 9E	2/2	move immediate date to serial I/O register	(S0)←data (S1)←data (S2)←data	
	EN SI	85	1/1	enable serial I/O interrupt		
	DIS SI	95	1/1	disable serial I/O interrupt		
	NOP	00	1/1	no operation		

Notes to Table 6.

1. PSW CY, AC affected
2. PSW CY affected
3. PSW PS affected
4. Execution of JTF and JNTF instructions resets the Timer Flag (TF).
5. PSW RBS affected
6. PSW SP_0, SP_1, SP_2 affected
7. (A) = 1111 P23, P22, P21, P20.
8. (S1) has a different meaning for read and write operation, see serial I/O interface.

* : 8, 9, A, B, C, D, E, F
● : 0, 2, 4, 6, 8, A, C, E
▲ : 1, 3, 5, 7, 9, B, D, F

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Supply voltage with respect to V_{SS}	V_{CC}	−0,5 to	+ 7 V
All input and output voltages with respect to V_{SS} (except for the port pins)	V_I, V_O	−0,5 to	+ 7 V
Voltage at any port pin with respect to V_{SS}; with R ⩾ 1 kΩ in series	V_I	−0,5 to	+ 12 V
Total power dissipation			
for SOT-117D	P_{tot}	max.	1 W
for SOT-135 and SOT-136A (SO-28)	P_{tot}	max.	0,6 W
Input output current	$\pm I_I$, I_O	max.	10 mA
Storage temperature range	T_{stg}	−65 to	+ 150 °C
Operating ambient temperature range	T_{amb}	0 to	+ 70 °C

D.C. CHARACTERISTICS

V_{SS} = 0 V; V_{CC} = 5 V (± 10%); T_{amb} = 0 to + 70 °C; all voltages with respect to V_{SS}; unless otherwise specified

	symbol	min.	typ.	max.		conditions
Supply voltage	V_{CC}	4,5	5,0	5,5	V	
Inputs						
Input voltage LOW all inputs except P23, SCLK	V_{IL}	−0,5	–	0,8	V	
Input voltage HIGH all inputs except XTAL1, P23, SCLK	V_{IH}	2,0	–	V_{CC}	V	
Input voltage HIGH XTAL1, P23, SCLK	V_{IH}	3,0	–	V_{CC}	V	
Input voltage LOW P23, SCLK	V_{IL}	−0,5	–	1,5	V	
Outputs						
Output voltage LOW	V_{OL}	–	–	0,45	V	I_{OL} = 1,6 mA
Output voltage LOW P10, P11	V_{OL}	–	–	2,5	V	I_{OL} = 7 mA
Output voltage LOW P23, SCLK	V_{OL}	–	–	0,45	V	I_{OL} = 5 mA
Output voltage HIGH all outputs unless open drain	V_{OH}	2,4	–	–	V	$-I_{OH}$ = 50 μA
Output leakage current open drain	$-I_{OL}$	–	–	10	μA	$V_{CC} \geqslant V_I \geqslant V_{SS}$

A.C. CHARACTERISTICS

V_{SS} = 0 V; V_{CC} = 5 V (± 10%); T_{amb} = 0 to + 70 °C; unless otherwise specified

	symbol	min.	typ.	max.		condition
Cycle time	t_{cy}	6,77	–	50	μs	4,43 MHz crystal = 6,77 μs