



Electronic
components
and materials

PHILIPS

MAB8400 section

Purchase of Philips' I²C components conveys a license under the Philips' I²C patent to use the components in the I²C-system provided the system conforms to the I²C specifications defined by Philips.



CONTENTS PAGE - MAB8400

1.0 DESCRIPTION

2.0 FEATURES

3.0 FUNCTIONAL DESCRIPTION

3.1 Program memory (ROM)

3.2 Data memory (RAM)

3.3 Input/Output

3.3.1 Parallel ports

3.3.2 Serial I/O

3.4 Interrupts

3.4.1 Interrupt logic

3.4.2 Interrupt program examples

3.5 Test input T1

3.6 High current outputs

3.7 Oscillator and clock

3.8 Timer/event counter

3.9 Program status word

3.10 Program counter

3.11 Central processing unit

3.12 Reset

3.13 Instruction set

ELECTRICAL SPECIFICATIONS

4.0 RATINGS

5.0 D.C. CHARACTERISTICS

6.0 A.C. CHARACTERISTICS

7.0 T1 ZERO CROSS CHARACTERISTICS

8.0 D.C. AND A.C. CHARACTERISTICS OF MAB8400Q BOND-OUT CHIP

8.1 $\overline{\text{HALT}}$ function

9.0 TIMING

9.1 Clock adjustment

9.2 Timing requirements for the P23 and SCLK input signals

9.3 Timing requirements for the P23 and SCLK output signals





10.0 PACKAGE OUTLINES

10.1 Pin designation MAB8410/20/40, MAB8400B

11.0 BOND-OUT VERSION MAB8400Q

11.1 Pin designation MAB8400Q

28-LEAD DUAL IN-LINE; PLASTIC (SOT-117D)

28-LEAD DUAL IN-LINE; CERAMIC (SOT-135)

28-LEAD FLAT-PACK; PLASTIC (SO-28; SOT-136A)

56-LEAD QUADRUPLÉ IN-LINE; PLASTIC (VO-35)



1.0 DESCRIPTION

The MAB8400 family of single-chip 8-bit microcomputers is manufactured in N-MOS. The family consists of 4 devices:

- o MAB8400B: NO ROM/128 RAM bytes ("Piggy-back version"),*
- o MAB8410 : 1K ROM/64 RAM bytes,
- o MAB8420 : 2K ROM/64 RAM bytes,
- o MAB8440 : 4K ROM/128 RAM bytes,

Each type has 20 quasi-bidirectional I/O port lines, two serial I/O lines, three single-level vectored interrupts, and an 8-bit timer/eventcounter 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.

*Note: The MAB8400Q 56 pin QUIL version is intended for emulation support on development systems only.

2.0 FEATURES

- o 8-bit: CPU, ROM, RAM, I/O in a single 28-lead DIL package
- o 1K, 2K or 4K ROM bytes
- o 64, 64 or 128 RAM bytes
- o 20 quasi-bidirectional I/O port lines
- o Two test inputs: one of which can be used to detect zero voltage cross-over, the other is also the external interrupt input
- o Single-level vectored interrupts: external, timer/event counter, serial I/O
- 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)
- o 8-bit programmable timer/event counter
- o Internal oscillator.
- o Over 80 instructions (based on MAB8048)
- o All instructions 1 or 2 cycles
- o Single 5 V supply ($\pm 10\%$)
- o Operating temperature range: 0 to +70°C for MAB versions, or -40 to +85°C for special MAF versions.

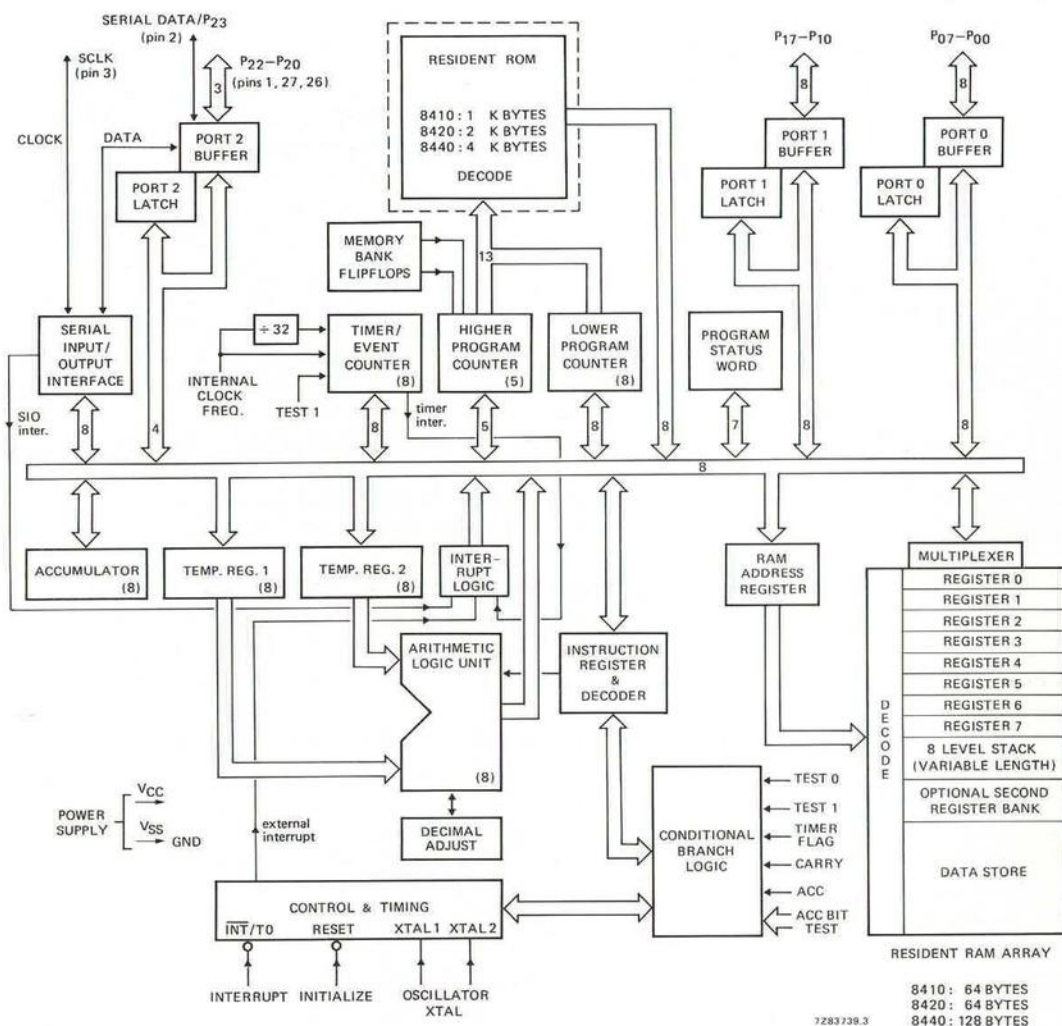


Fig.1a Block diagram of the MAB8400 family.

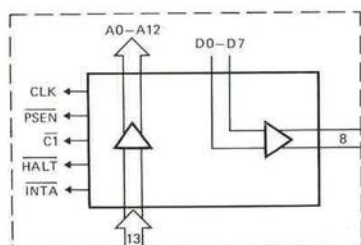


Fig.1b Replacement of dotted part in Fig.1a, showing the MAB8400Q bond-out version.

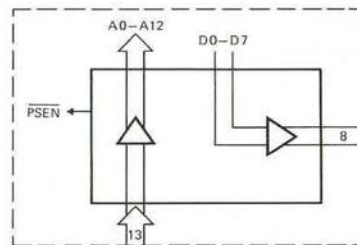


Fig.1c Replacement of dotted part in Fig.1a for MAB8400B ('Piggy-back').

3.0 FUNCTIONAL DESCRIPTION

3.1 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. Fig.2 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 initialised (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 organised 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, if a memory bank change is wanted, the required bank must be selected prior to a CALL or JMP instruction.

3.2 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. Fig.3 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 RBO 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 designed as an 8-level program counter stack (2 locations per level), or as general purpose RAM. The program counter stack (Fig.4) 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.

The stack pointer, when initialised 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), 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.

Nesting of subroutines within subroutines can continue up to 8 times without overflowing the stack. If an overflow does occur, the deepest address (location 8 and 9) will be overwritten and lost since the stack pointer overflows from 111 to 000. It also underflows from 000 to 111.

The value of the saved contents of the program counter is different for an interrupt CALL compared to a normal CALL to subroutine. With an interrupt CALL, the program counter return address is saved; with a subroutine CALL, the saved program counter value is one less than the program counter return address.

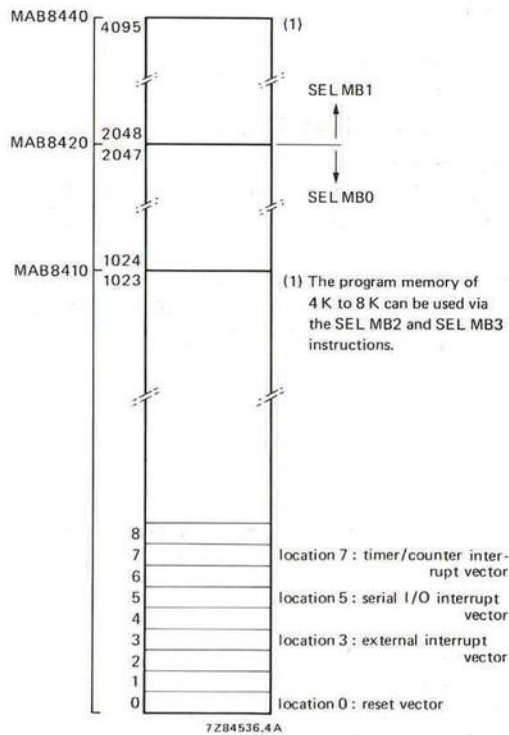


Fig.2 Program memory map.

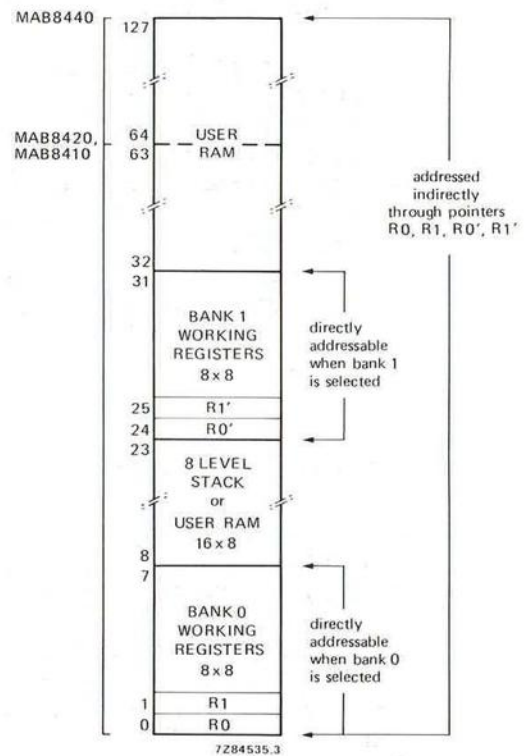


Fig.3 Data memory map.

7Z89147.2

STACK POINTER		
1 1 1		R23
		22
1 1 0		21
		20
1 0 1		19
		18
1 0 0		17
		16
0 1 1		15
		14
0 1 0		13
		12
0 0 1		11
		10
0 0 0	PC ₇ PC ₆ PC ₅ PC ₄ PC ₃ PC ₂ PC ₁ PC ₀	9
	PSW ₇ PSW ₆ PC ₁₂ PSW ₄ PC ₁₁ PC ₁₀ PC ₉ PC ₈	R8
	MSB	LSB

Fig.4 Program counter stack.

3.3 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 (INT/T0): when used as a test input it can be tested by the conditional branch instructions JTO 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.

3.3.1 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. Fig.6(a) 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 ($\approx 50 \text{ k}\Omega$). 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 high impedance pull-up. 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.

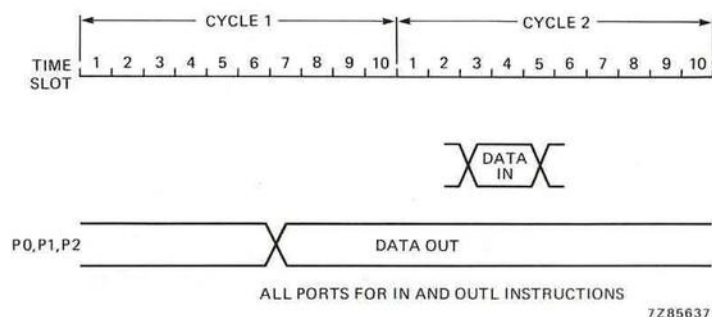


Fig. 5 Timing diagram of all ports on IN and OUTL instructions; for ANL and ORL instructions, the ports change on the time slot 7 of cycle 2.

Two other interface output configuration can also be specified.

- open drain output with pull-up (Fig.6(b))
- open drain output without pull-up (Fig.6(c))
(P23 only with open drain configuration).

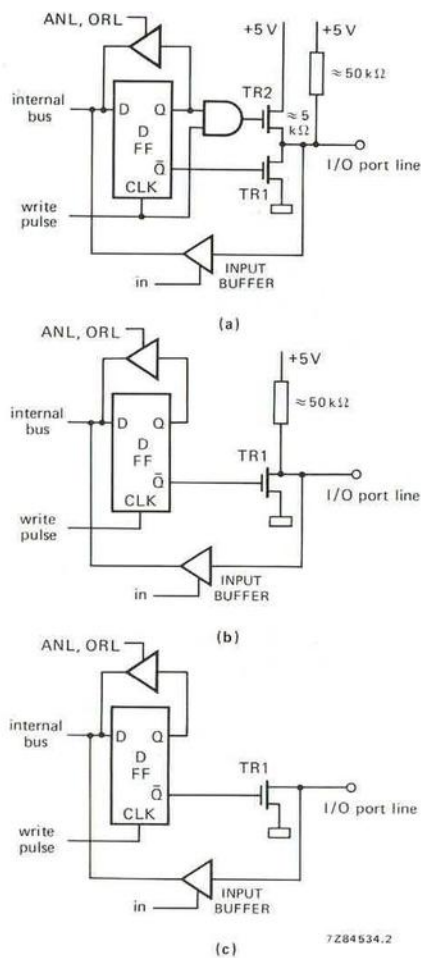


Fig.6 Quasi-bidirectional I/O interface with (a) push-pull output with pull-up resistance; (b) open drain output with pull-up resistance; (c) open drain output without pull-up resistance.



3.3.2 Serial I/O

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. Such a system saves significant processing time and memory space compared with a conventional microcomputer employing a software serial interface.

When the addressing facility is not required, for instance in a system with only two microcomputers on the serial bus, 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.

A more detailed description of serial I/O is contained in a separate section of this manual.

3.4 Interrupt

When the external interrupt is enabled, a HIGH to LOW transition on the $\overline{\text{INT}}/\text{T0}$ input initiates an external interrupt subroutine which causes a call to program memory location 3 following completion of the current instruction. 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.

External interrupts are always latched, even when the external interrupt is disabled. 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. When a second interrupt occurs during the running of the first routine the computer will enter the second routine after having executed one instruction in the main program. If 2 or 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 the timer/counter interrupt vector location 7.

3.4.1 Interrupt logic

The external interrupt of the MAB8400 is active on the negative edge; a HIGH to LOW transition on the $\overline{\text{INT}}/\text{T0}$ input will be latched in a 'digital filter/latch' (3-bit shift register in which the negative interrupt is stored; see interrupt logic diagram Fig.7) and, when enabled, initiates an external interrupt service routine. The interrupt activity remains latched in the 'digital filter' until it is taken over by the 'External Interrupt Flag', which in turn resets the digital filter. This flag can only be set when external interrupts are enabled.

The 'External Interrupt Flag', when set, causes the current instruction to end and forces a subroutine CALL to program memory location 3. During this forced CALL the 'External Interrupt Flag' is reset, which enables new interrupts to be latched in the 'digital filter/latch'. Also the 'Interrupt in Progress Flag' is set, which causes other interrupts to be ignored and latched until the RETR instruction is executed. These (latched) interrupts will start an interrupt routine after the program has returned to the main part.

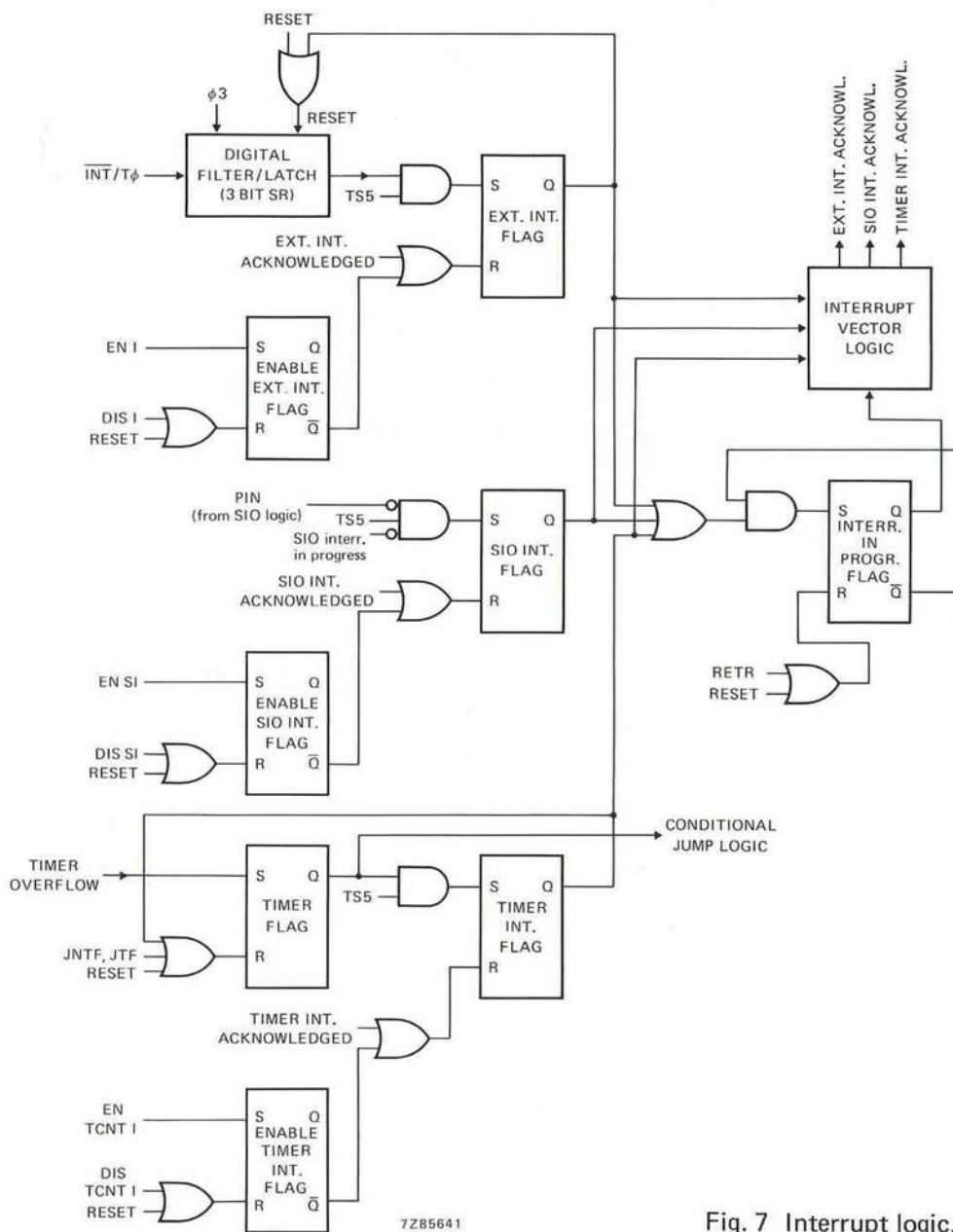


Fig. 7 Interrupt logic.

Note:

1. $\overline{\text{INT}}/\text{T0}$ negative edge is *a/ways* latched in digital filter latches.
2. Proper interrupt timing is ensured when $\overline{\text{INT}}/\text{T0}$ is HIGH for $> 1,33$ time slot followed by LOW for $> 2,33$ time slot.
3. A DIS I instruction within an external interrupt service routine clears a pending external interrupt.
4. When interrupt in progress flag is set, further interrupts are latched but ignored, until RETR is executed.

3.4.2 Interrupt program examples

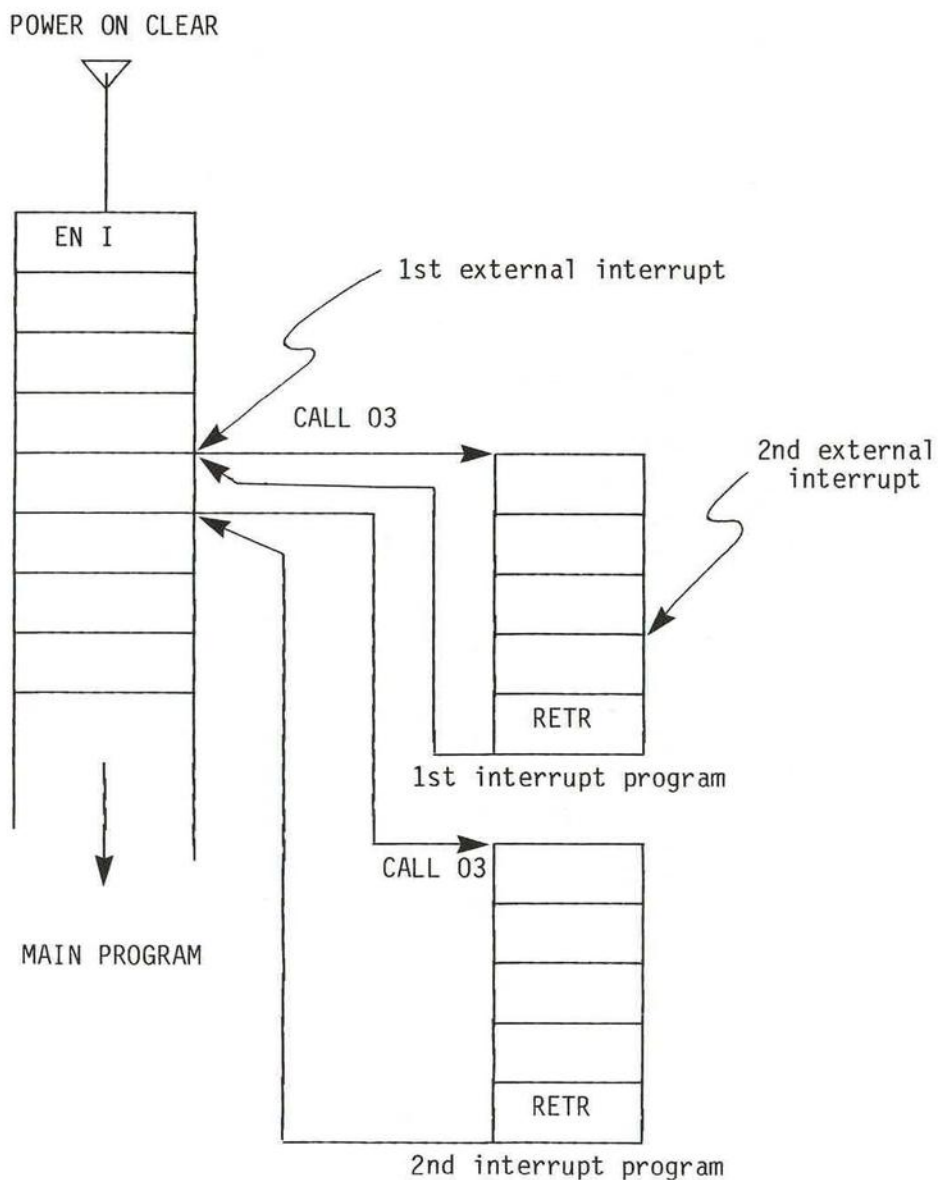


Fig.8 An external interrupt during the first interrupt program remains latched (pending) until this interrupt program is complete.

3.4.2 Interrupt program examples (continued)

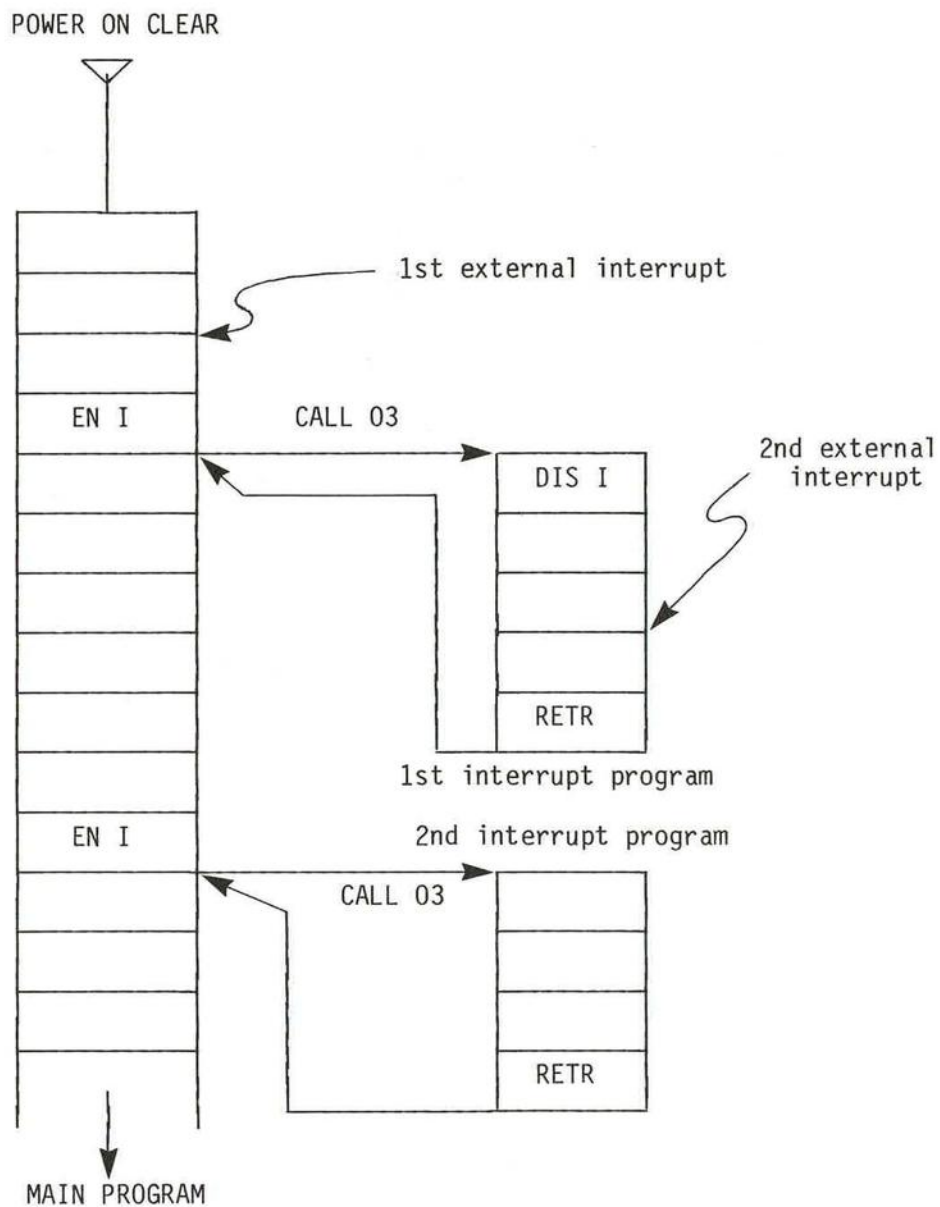


Fig.9 Second external interrupt is postponed until enabled again.

3.4.2 Interrupt program examples (continued)

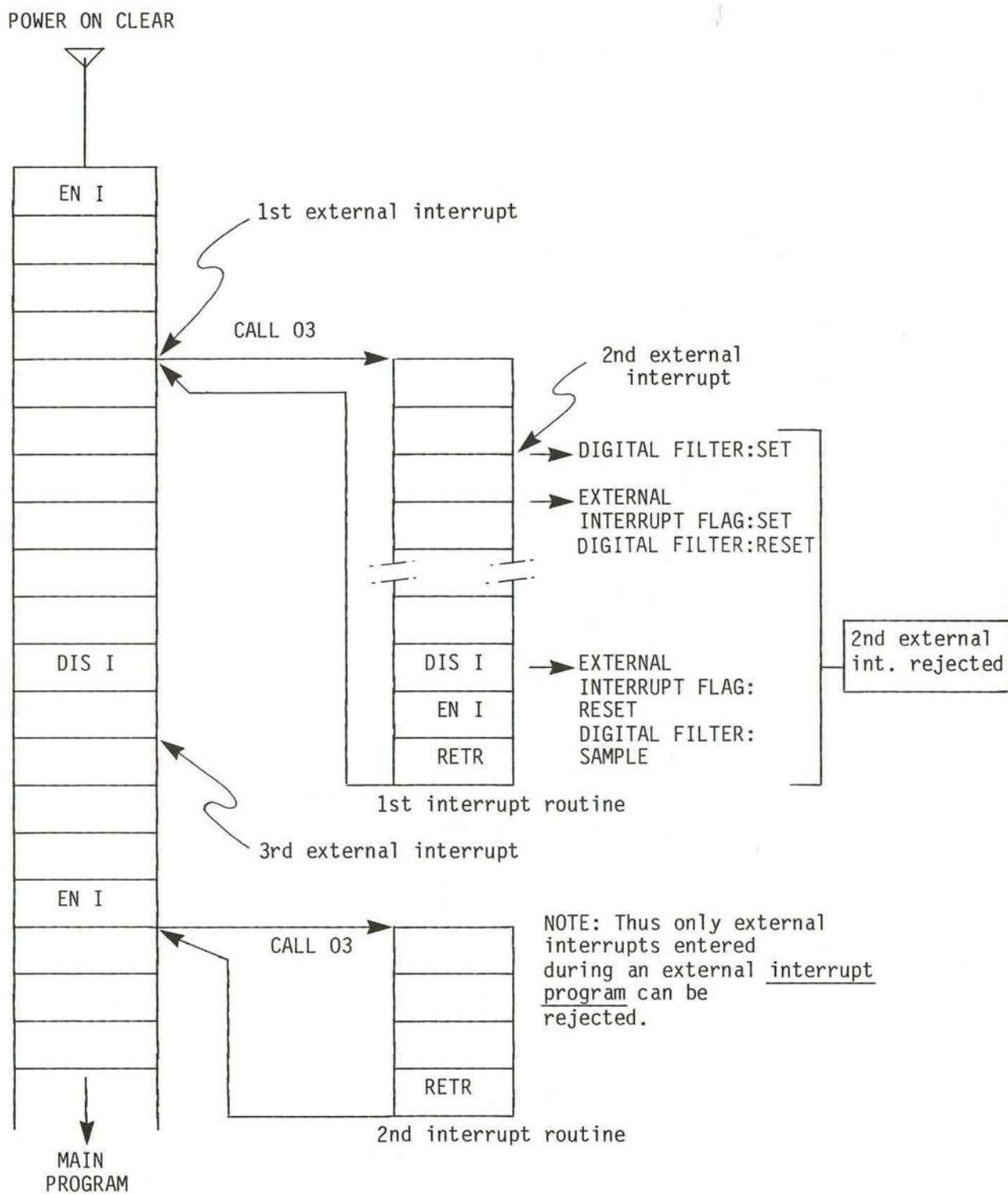


Fig.10 Clearing of previous external interrupts.

3.4.2 Interrupt program examples (continued)

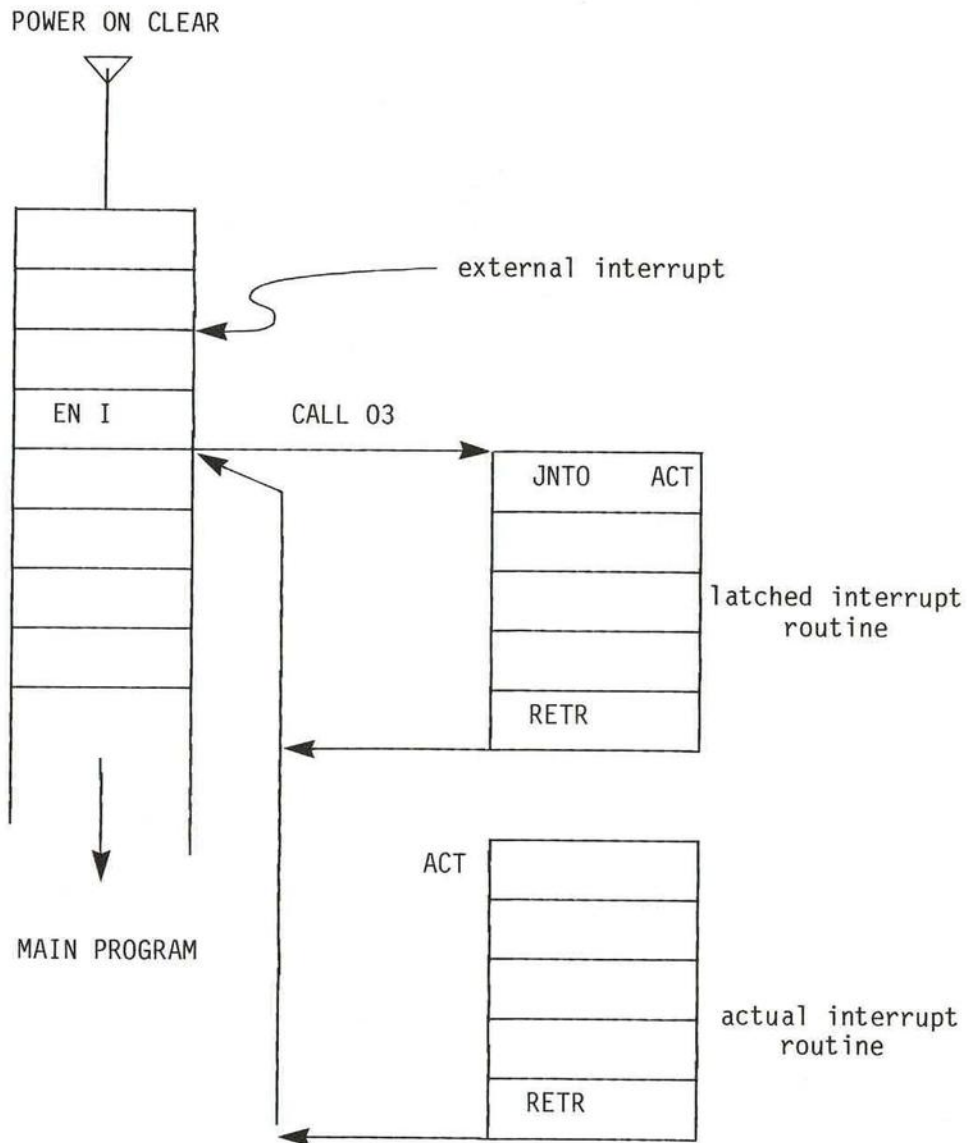


Fig.11 Detection of previous and actual external interrupt.

For proper latching in the 'digital filter/latch', the external interrupt should be HIGH for at least 1,33 time slots and LOW for at least 2,33 time slots (1 time slot = 3 clock cycles). The digital filter, therefore, acts as a noise suppression filter.

The maximum rate at which interrupt edges can be repeated and latched is 4 machine cycles.

Note that $\overline{\text{INT}}/\text{T0}$ edges are always latched, even when the external interrupts are disabled.

The SIO interrupt circuit is similar to the external interrupt circuit. Here the 'PIN' bit (SIO status word 1) can initiate service routines.

Timer interrupt routines can be started by the overflow of the timer, causing the timer flag to be set. This, when allowed, also sets the 'Timer Interrupt Flag', which then in turn resets the timer flag.

3.5 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 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 is 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 in section 3.8.

3.6 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 paralleled (if their logic outputs are always the same) to give 14 mA drive at 2,5 V.

3.7 Oscillator and clock

A crystal, ceramic resonator, inductor or resistor connected between XTAL1 and XTAL2 (see Fig.12) 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 μ s machine cycle.

The MAB8400 family has dynamic logic; for adequate refreshing the oscillator frequency must be at least 1 MHz.

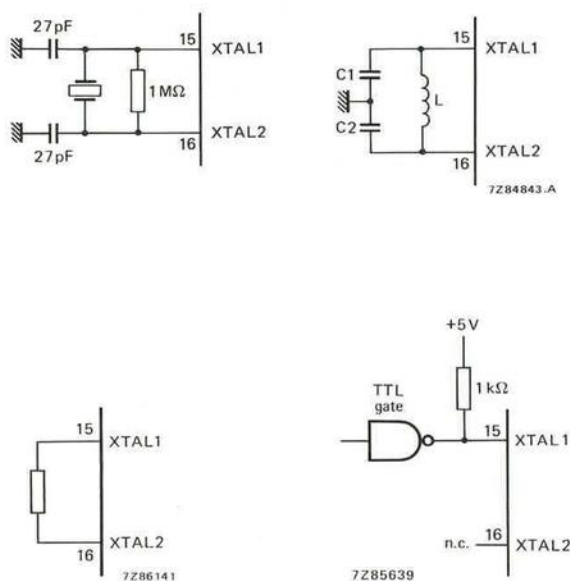


Fig.12 Connections for clock adjustment.

For a detailed specification on clock adjustment see section 9.1.

3.8 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.13). Table 1 gives the instructions that control the counter and the prescaler and the functions performed.

Table 1 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.

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 μ s 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 and JNTF - instruction. Overflow also generates an interrupt to the processor when the timer/event counter interrupt is enabled.

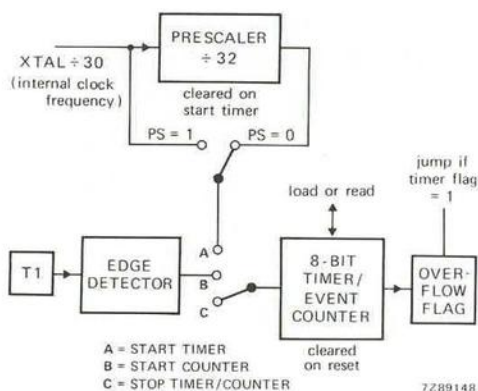


Fig.13 Timer/event counter.

3.9 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.14). 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.

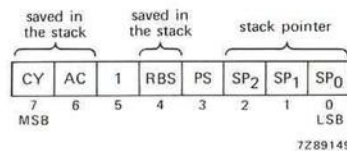
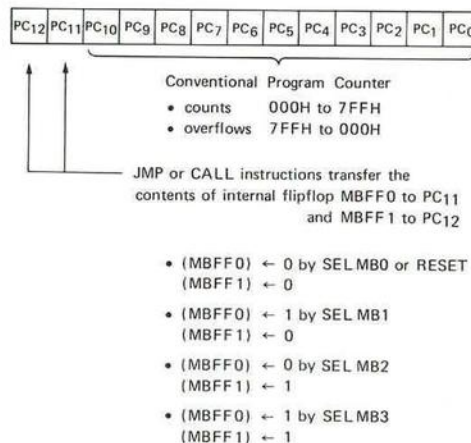


Fig.14 Program status word.

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 the case of an interrupt. Bits 7, 6 and 4 are stored in the program counter stack during subroutines 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.

3.10 Program counter

A 13-bit program counter is used so that up to 8K bytes of ROM can be addressed. Fig.15 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.



7289150

Fig.15 Program counter.

3.11 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 2 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 the address of which are pointed to by the accumulator.

Table 2 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

TABLE 3. 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															
1	INC arr															
2	XCH A, arr															
3	XCHD A, arr															
4	ORL A, arr															
5	ANL A, arr															
6	ADD A, arr															
7	ADDC A, arr															
8																
9																
A	MOV arr, A															
B	MOV arr, data															
C	DEC arr															
D	XRL A, arr															
E	DJNZ arr, addr															
F	MOV A, arr															

Table 4 shows the MAB8400 family instructions (including the five instructions for serial I/O operation) absent from the MAB8048 instruction set.

Table 4 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 5 shows the MAB8048 instructions absent from the MAB8400 family instruction set.

Table 5 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	* JN1 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 6 Instruction set is shown on the next 5 pages.

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

ACCUMULATOR

mnemonic	opcode (hex.)	bytes/ cycles	description	function	notes
ACCUMULATOR (cont.)					
RLC A	F7	1/1	rotate A left through carry	$(A_n + 1) \leftarrow A_n$ $(A_0) \leftarrow (C), (C) \leftarrow (A_7)$	2 n = 0-6
RR A	77	1/1	rotate A right	$(A_n) \leftarrow (A_n + 1)$ $(A_7) \leftarrow (A_0)$	2 n = 0-6
RRC A	67	1/1	rotate A right through carry	$(A_n) \leftarrow (A_n + 1)$ $(A_7) \leftarrow (C), (C) \leftarrow (A_0)$	2 n = 0-6
DA A	57	1/1	decimal adjust A		2
SWAP A	47	1/1	swap nibbles of A	$(A_{4-7}) \leftrightarrow (A_{0-3})$	2
DATA MOVES					
MOV A, Rr	F*	1/1	move register contents to A	$(A) \leftarrow (Rr)$	r = 0-7
MOV A, @Rr	F0 F1	1/1	move RAM data, addressed by Rr, to A	$(A) \leftarrow ((R0))$ $(A) \leftarrow ((R1))$	
MOV A, #data	23 data	2/2	move immediate data to A	$(A) \leftarrow \text{data}$	
MOV Rr, A	A*	1/1	move accumulator contents to register	$(Rr) \leftarrow (A)$	r = 0-7
MOV @Rr, A	A0 A1	1/1	move accumulator contents to RAM location addressed by Rr	$((R0)) \leftarrow (A)$ $((R1)) \leftarrow (A)$	
MOV Rr, #data	B* data	2/2	move immediate data to Rr	$(Rr) \leftarrow \text{data}$	
MOV @Rr, #data	B0 data B1 data	2/2	move immediate data to RAM location addressed by Rr	$((R0)) \leftarrow \text{data}$ $((R1)) \leftarrow \text{data}$	r = 0-7
XCH A, Rr	2*	1/1	exchange accumulator contents with Rr	$(A) \leftrightarrow (Rr)$	
XCH A, @Rr	20 21	1/1	exchange accumulator contents with RAM data addressed by Rr	$(A) \leftrightarrow ((R0))$ $(A) \leftrightarrow ((R1))$	
XCHD A, @Rr	30 31	1/1	exchange lower nibbles of A and RAM data addressed by Rr	$(A_{0-3}) \leftrightarrow ((R0_{0-3}))$ $(A_{0-3}) \leftrightarrow ((R1_{0-3}))$	3
MOV A, PSW	C7	1/1	move PSW contents to accumulator	$(A) \leftarrow (\text{PSW})$	
MOV PSW, A	D7	1/1	move accumulator bit 3 to PSW ₃	$(\text{PSW}_3) \leftarrow (A_3)$	
MOV P, A	A3	1/2	move indirectly addressed data in current page to A	$(PC_{0-7}) \leftarrow (A), (A) \leftarrow ((PC))$	

FLAGS	CLR C CPL C	97 A7	1/1 1/1	clear carry bit complement carry bit	(C) ← 0 (C) ← NOT(C)	2 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	
	DEC @Rr	C0 C1	1/1	decrement RAM data, addressed by Rr, by 1	((R0)) ← ((R0)) - 1 ((R1)) ← ((R1)) - 1	
	JMP addr	● 4 address	2/2	unconditional jump within a 2Kbank	(PC8-10) ← addr8-10 (PC0-7) ← addr0-7 (PC11-12) ← MBFF 0-1 (PC0-7) ← ((A))	
BRANCH	JMPP @A	B3	1/2	indirect jump within a page	(Rr) ← (Rr) - 1	r = 0-7
	DJNZ Rr, addr	E* address	2/2	decrement Rr by 1 and jump if not zero to addr	if (Rr) not zero (PC0-7) ← addr	
	DJNZ @Rr, addr	E0 E1	2/2	decrement RAM data, addressed by Rr, by 1 and jump if not zero to addr	((R0)) ← ((R0)) - 1 if ((R0)) not zero (PC0-7) ← addr ((R1)) ← ((R1)) - 1 if ((R1)) not zero (PC0-7) ← addr	
	JBb addr	▲ 2 address	2/2	jump to addr if Acc. bit b = 1	if b = 1: (PC0-7) ← addr	
	JC addr	F6 address	2/2	jump to addr if C = 1	if C = 1: (PC0-7) ← addr	
	JNC addr	E6 address	2/2	jump to addr if C = 0	if C = 0: (PC0-7) ← addr	b = 0-7
	JZ addr	C6 address	2/2	jump to addr if A = 0	if A = 0: (PC0-7) ← addr	
	JNZ addr	96 address	2/2	jump to addr if A is NOT zero	if A ≠ 0: (PC0-7) ← addr	
	JT0 addr	36 address	2/2	jump to addr if T0 = 1	if T0 = 1: (PC0-7) ← addr	
	JNT0 addr	26 address	2/2	jump to addr if T0 = 0	if T0 = 0: (PC0-7) ← addr	
	JT1 addr	56 address	2/2	jump to addr if T1 = 1	if T1 = 1: (PC0-7) ← addr	
	JNT1 addr	46 address	2/2	jump to addr if T1 = 0	if T1 = 0: (PC0-7) ← addr	
	JTF addr	16 address	2/2	jump to addr if Timer Flag = 1	if TF = 1: (PC0-7) ← addr	
	JNTF addr	06 address	2/2	jump to addr if Timer Flag = 0	if TF = 0: (PC0-7) ← addr	
						4

	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 ₀₋₁₀ (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, Sn	0C 0D	1/2	move serial I/O register contents to accumulator	(A)←(S0) (A)←(S1)	8
	MOV Sn, A	3C 3D 3E	1/2	move accumulator contents to serial I/O register	(S0)←(A) (S1)←(A) (S2)←(A)	
	MOV Sn, #data	9C 9D 9E	2/2	move immediate data 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 SP0, SP1, SP2 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

MAB8400 electrical specifications

4.0 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 $R \geq 1 \text{ k}\Omega$ 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

5.0 D.C. CHARACTERISTICS

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

parameter	symbol	min.	typ.	max.	unit	conditions
Supply voltage	V_{CC}	4,5	5,0	5,5	V	
Supply current	I_{CC}^*	65	80	95	mA	MAB8400B;Q
	I_{CC}^*	55	70	85	mA	MAB8410/20/40
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	$I_{OL} = 1,6 \text{ mA}$
Output voltage LOW P10, P11	V_{OL}	—	—	2,5	V	$I_{OL} = 7 \text{ mA}$
Output voltage LOW P23, SCLK	V_{OL}	—	—	0,45	V	$I_{OL} = 5 \text{ mA}$
Output voltage HIGH all outputs unless open drain	V_{OH}	2,4	—	—	V	$-I_{OH} = 50 \mu\text{A}$
Output leakage current open drain	$-I_{OL}$	—	—	10	μA	$V_{CC} \geq V_I \geq V_{SS}$

* These values are not guaranteed; they are derived from development sample data and are subject to change.

6.0 A.C. CHARACTERISTICS
 $V_{SS} = 0\text{ V}$; $V_{CC} = 5\text{ V}$ ($\pm 10\%$); $T_{amb} = 0\text{ to } +70\text{ }^{\circ}\text{C}$; unless otherwise specified

parameter	symbol	min.	typ.	max.	unit	condition
Cycle time	t_{CY}	6,77	—	50	μs	$\left\{ \begin{array}{l} 4,43\text{ MHz crystal} \\ = 6,77\text{ }\mu\text{s} \end{array} \right.$

7.0 T1 ZERO-CROSS CHARACTERISTICS
 $T_{amb} = 0\text{ to } +70\text{ }^{\circ}\text{C}$; $V_{CC} = 5,5\text{ V} \pm 1\text{ V}$; $V_{SS} = 0\text{ V}$; $C_L = 80\text{ pF}$

parameter	symbol	min.	max.	unit	test conditions
Zero-cross detection input (T1) peak-to-peak value	$V_{ZX(p-p)}$	1	3	V	a.c. coupled, $C = 0,2\text{ }\mu\text{F}$
Zero-cross accuracy	A_{ZX}	—	± 135	mV	50 Hz sine-wave
Zero-cross detection input frequency (T1)	F_{ZX}	0,05	1	kHz	

8.0 D.C. AND A.C. CHARACTERISTICS OF MAB8400Q BOND-OUT CHIP

The non-standard outputs can drive at least 1 standard LS-TTL input, this means:

$$V_{OLmax} = 0,4\text{ V at } I_{OL} = 0,4\text{ mA } (V_{CC} = V_{CCmin}).$$

$$V_{OHmin} = 2,4\text{ V at } -I_{OH} = 50\text{ }\mu\text{A } (V_{CC} = V_{CCmin}).$$

The inputs are identical to the standard MAB8400 inputs.

The a.c. characteristics are listed in the table below.

 $T_{amb} = 0\text{ to } +70\text{ }^{\circ}\text{C}$; $V_{CC} = 5\text{ V} \pm 10\%$; $V_{SS} = 0\text{ V}$; $f_{XTAL} = 4,43\text{ MHz}$.

parameter	symbol	min.	max.	unit	remarks
Cycle time	t_{CY}	6,77	30	μs	$t_{CY} = 30\text{CP}$
Control pulse width	t_{CC}	2,03	—	μs	$t_{CC} = 9\text{CP}$
Address to $\overline{\text{PSEN}}$ set-up	t_{AS}	0	226	ns	$t_{AS} \leq 1\text{CP}$
Data to $\overline{\text{PSEN}}$ set-up	t_{DS}	346	—	ns	$t_{DS} = 1\text{CP} + 120\text{ ns}$
Data hold time	t_{DR}	0	—	ns	
Address to Data-in	t_{AD}	—	1,91	μs	$t_{AD} = 9\text{CP} + t_{AS} - t_{DS}$
Time from $\overline{\text{PSEN}}$ to $\overline{\text{C1}}$	t_{PC}	677	—	ns	$t_{PC} = 3\text{CP}$
Time from $\overline{\text{INTA L}}$ to $\overline{\text{PSEN}}$	t_{IP0}	677	—	ns	$t_{IP0} = 3\text{CP}$
Time from $\overline{\text{INTA H}}$ to $\overline{\text{PSEN}}$	t_{IP1}	1,35	—	μs	$t_{IP1} = 6\text{CP}$
Halt set-up to $\overline{\text{PSEN}}$	t_{HS}	3,39	—	μs	$t_{HS} = 15\text{CP}$
Halt hold time from $\overline{\text{PSEN}}$	t_{HH}	678	—	ns	$t_{HH} = 3\text{CP}$

NOTE: CP = clock pulse.

8.1 $\overline{\text{HALT}}$ function

With the $\overline{\text{HALT}}$ input the processor can be stepped through the program, one instruction at a time.

While stopped, the address of the next instruction to be fetched is available on the address bus.

The user can therefore follow the program through each of the instruction steps. Fig.19 is a timing diagram, showing the $\overline{\text{HALT}}$ function.

The MAB8400Q operates in a $\overline{\text{HALT}}$ mode as follows:

1. The processor is requested to stop by applying a low level on $\overline{\text{HALT}}$. The processor senses this input continuously: it must be active low during at least TS4-TS9.
2. If the $\overline{\text{HALT}}$ line is found active, the processor finishes its current instruction and inserts a NOP opcode in the instruction register. The program counter and timer/ctr are no longer updated. The address of the next instruction is present on the address-bus.
3. Each following cycle that the $\overline{\text{HALT}}$ line is sampled active, a NOP is inserted in the instruction register (IR).
4. The processor continues if the $\overline{\text{HALT}}$ input is found HIGH during TS4-TS9. Instead of NOP, the next instruction is fetched and inserted in IR.

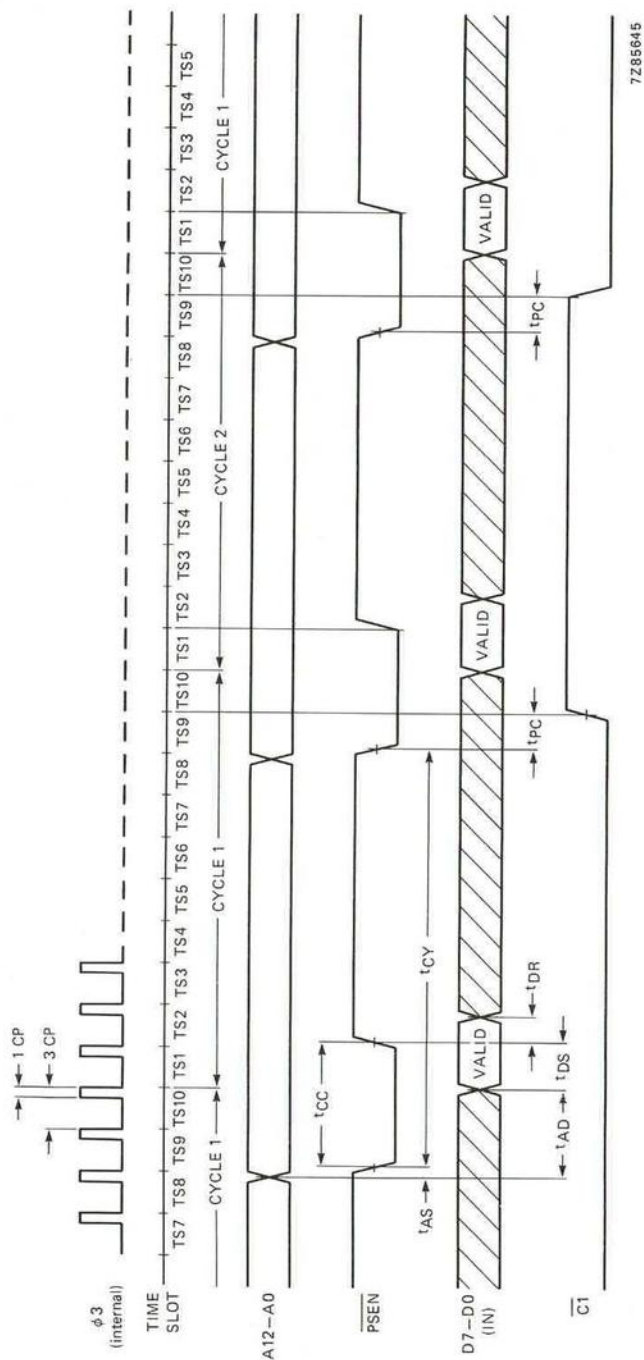


Fig.18 Memory-access timing MAB8400Q.

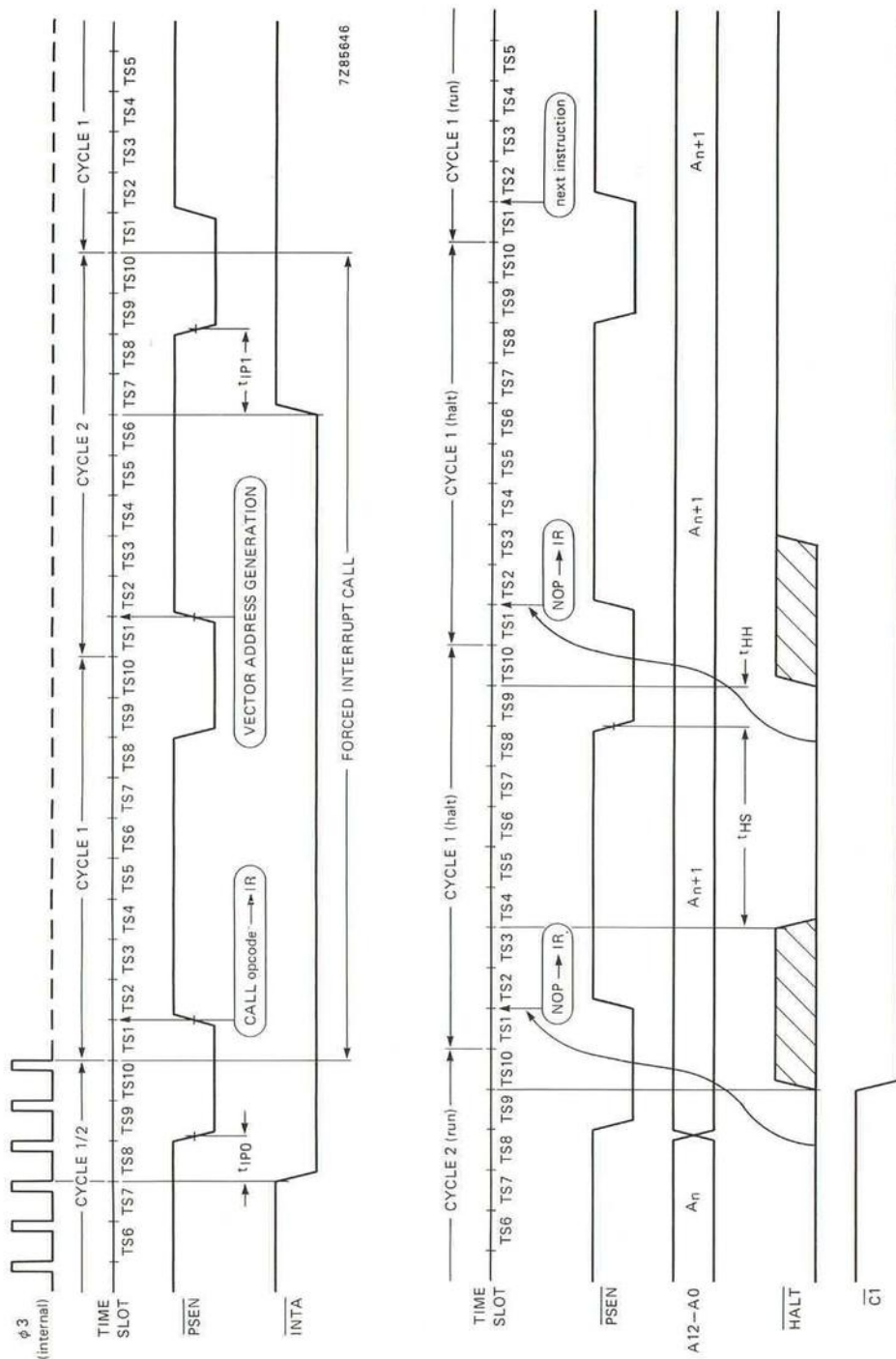
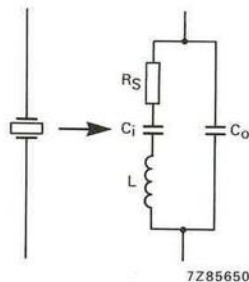
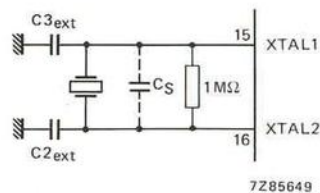


Fig.19 INTA and HALT timing MAB8400Q.

9.0 TIMING

9.1 Clock adjustment

Quartz crystal adjustment



Quartz crystal equivalent.

Quartz crystal parameter

Cut, resonance frequency: AT, 1 to 6 MHz (fundamental)

Dynamic capacitance (C₁): ≤10 pF

Parallel capacitance: see below (C_o + C_s)

Series resistance: ≤60 Ω

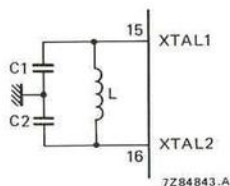
Oscillator circuit

	min.	max.
External + wiring capacitance on XTAL1: C _{3_ext}	27	70 pF
External + wiring capacitance on XTAL2: C _{2_ext}	27	70 pF
Capacitance in parallel with crystal: C _o + C _s	—	2C _L

$$\text{where: } C_L = \left(\frac{1}{C_{3_ext} + C_i} + \frac{1}{C_{2_ext} + C_o} \right)^{-1}$$

$$C_i = C_o \approx 25 \text{ pF}$$

LC adjustment



frequency	C1 = C2	L
3,0 MHz	33 pF	100 μH
4,0 MHz	33 pF	56 μH
4,4 MHz	33 pF	47 μH

With C1 = C2 and C = C1 + 27 pF the following formula is applicable:

$$f = \frac{1,46}{2\pi\sqrt{LC}}$$

9.2 Timing requirements for the P23 and SCLK input signals

t_{BUF}	$\geq 14t_{\text{XTAL}}$
$t_{\text{HD;STA}}$	$\geq 14t_{\text{XTAL}}$
t_{HIGH}	$\geq 17t_{\text{XTAL}}$
t_{LOW}	$\geq 17t_{\text{XTAL}}$
$t_{\text{SU;STO}}$	$\geq 14t_{\text{XTAL}}$
$t_{\text{HD;DAT}}$	> 0
$t_{\text{SU;DAT}}$	$\geq 250 \text{ ns}$
t_{RD}	$\leq 1 \mu\text{s}$
t_{RC}	$\leq 1 \mu\text{s}$
t_{FD}	$\leq 1 \mu\text{s}$
t_{FC}	$\leq 0,3 \mu\text{s}$

t_{XTAL} = one period of the XTAL input frequency (f_{XTAL}) = 226 ns for f_{XTAL} = 4,43 MHz.

These figures apply to all modes.

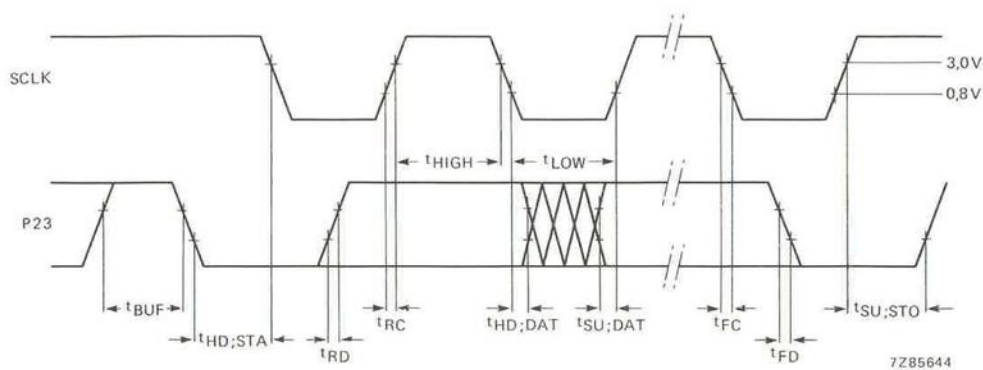


Fig.20 Input timing.

9.3 Timing requirements for the P23 and SCLK output signals

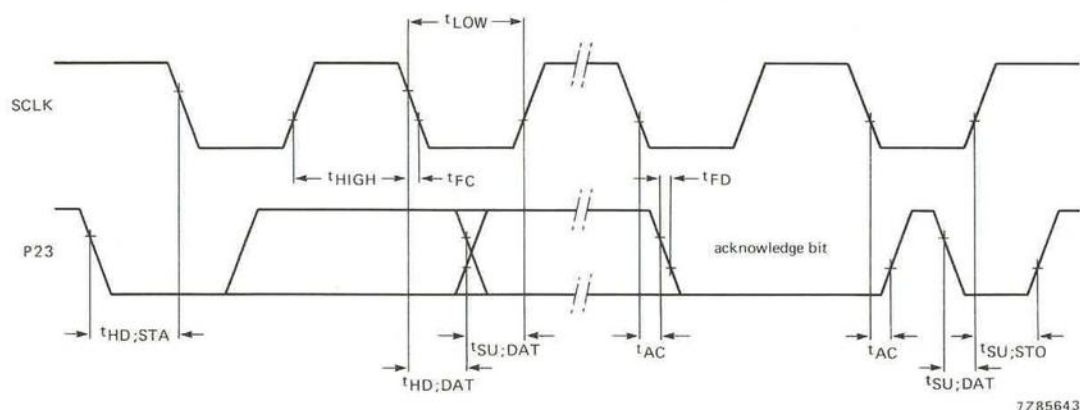


Fig.21 Output timing.

	high-speed mode (ASC in S2 = 0)	low-speed mode (ASC in S2 = 1)
$t_{SU;STA}$	$\frac{1}{2} (DF + 9) t_{XTAL}$	$\frac{3}{4} (DF + 9) t_{XTAL}$
t_{HIGH}	$\frac{1}{2} (DF + 3) t_{XTAL}$	$\frac{3}{4} (DF + 3) t_{XTAL}$
t_{LOW}	$\frac{1}{2} (DF - 3) t_{XTAL}$	$\frac{1}{4} (DF - 3) t_{XTAL}$
$t_{SU;STO}$	$\frac{1}{2} (DF - 3) t_{XTAL}$	$\frac{1}{4} (DF - 3) t_{XTAL}$
$t_{HD;DAT}$ (only for SLAVE TRANSMITTER)	$\geq 6 t_{XTAL}$ $\leq 12 t_{XTAL}$	$\geq 6 t_{XTAL}$ $\leq 12 t_{XTAL}$
$t_{SU;DAT}$ (only for MASTER TRANSMITTER)	$6 t_{XTAL}$	$6 t_{XTAL}$
t_{AC}	$\geq 6 t_{XTAL}$ $\leq 12 t_{XTAL}$	$\geq 6 t_{XTAL}$ $\leq 12 t_{XTAL}$
t_{FD}, t_{FC}	$\leq 100 \text{ ns at } C_b = 400 \text{ pF}^*$	$\leq 100 \text{ ns at } C_b = 400 \text{ pF}^*$

t_{XTAL} = one period of the XTAL input frequency (f_{XTAL}) = 226 ns for f_{XTAL} = 4.43 MHz.
DF = dividing factor.

* C_b is maximum bus capacitance for each line.

10.0 PACKAGE OUTLINES

MAB8400B: 28-lead "Piggy-back" package (with up to 28-lead EPROM on top)

MAB8400Q: 56-lead QUIL; plastic (V0-35)

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 (S0-28; SOT-136A)

10.1 PIN DESIGNATION MAB8410/20/40, MAB8400B

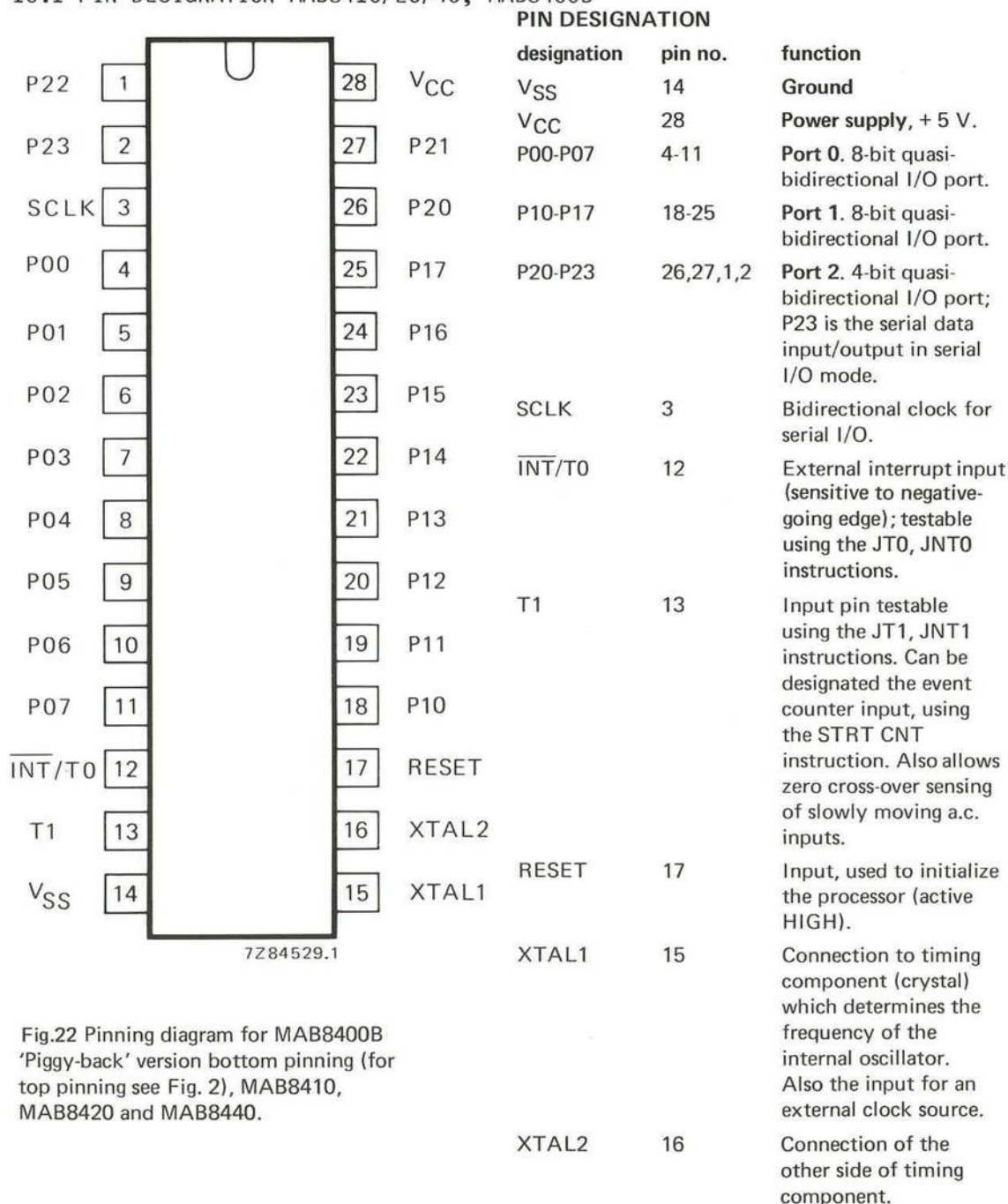
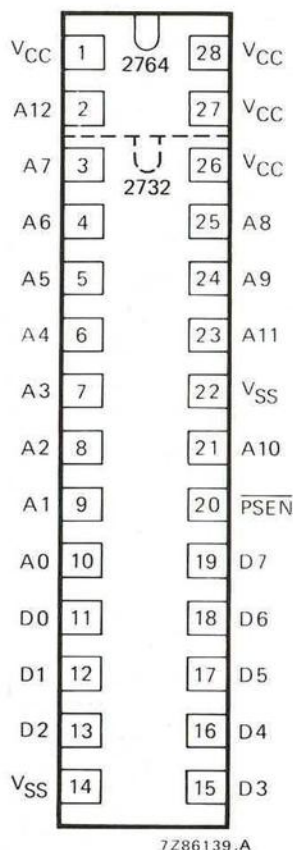


Fig.22 Pinning diagram for MAB8400B 'Piggy-back' version bottom pinning (for top pinning see Fig. 2), MAB8410, MAB8420 and MAB8440.

10.1 PIN DESIGNATION MAB8410/20/40, MAB8400B (continued)



PIN DESIGNATION

designation	pin no.	function
VSS	14, 22	Ground
VCC	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
PSEN	20	Program store enable

Fig.23 Pinning diagram for MAB8400B 'Piggy-back' version top pinning (for bottom pinning see Fig. 22); to access a 2732 or 2764 EPROM.

- Note :
- 1) A 2732 EPROM has to be inserted with its pin 1 in the pin 3 location of the MAB8400B top pinning.
 - 2) A 2716 EPROM can be used when its Vpp pin (pin 23 on MAB8400B top pinning) is disconnected from A11 and connected to VCC.
 - 3) Required access-time EPROM: < 2μs @ 4,43 MHz XTAL.

11.0 BOND-OUT VERSION MAB8400Q

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

11.1 PIN DESIGNATION MAB8400Q

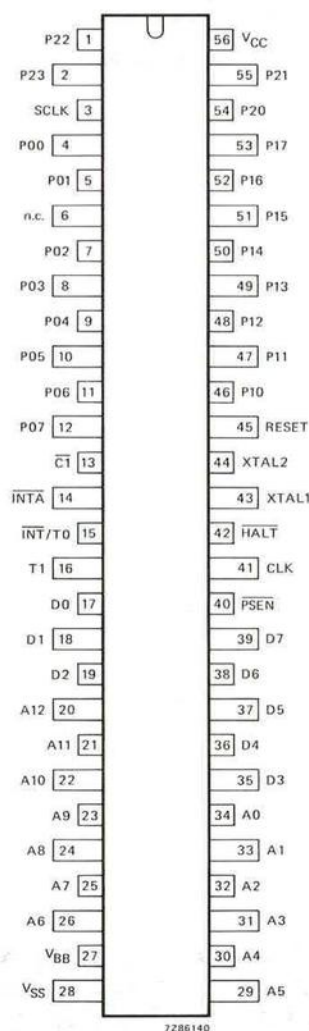


Fig.24 Pinning diagram for MAB8400Q; 56-leads, quadrate in-line.

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.
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.

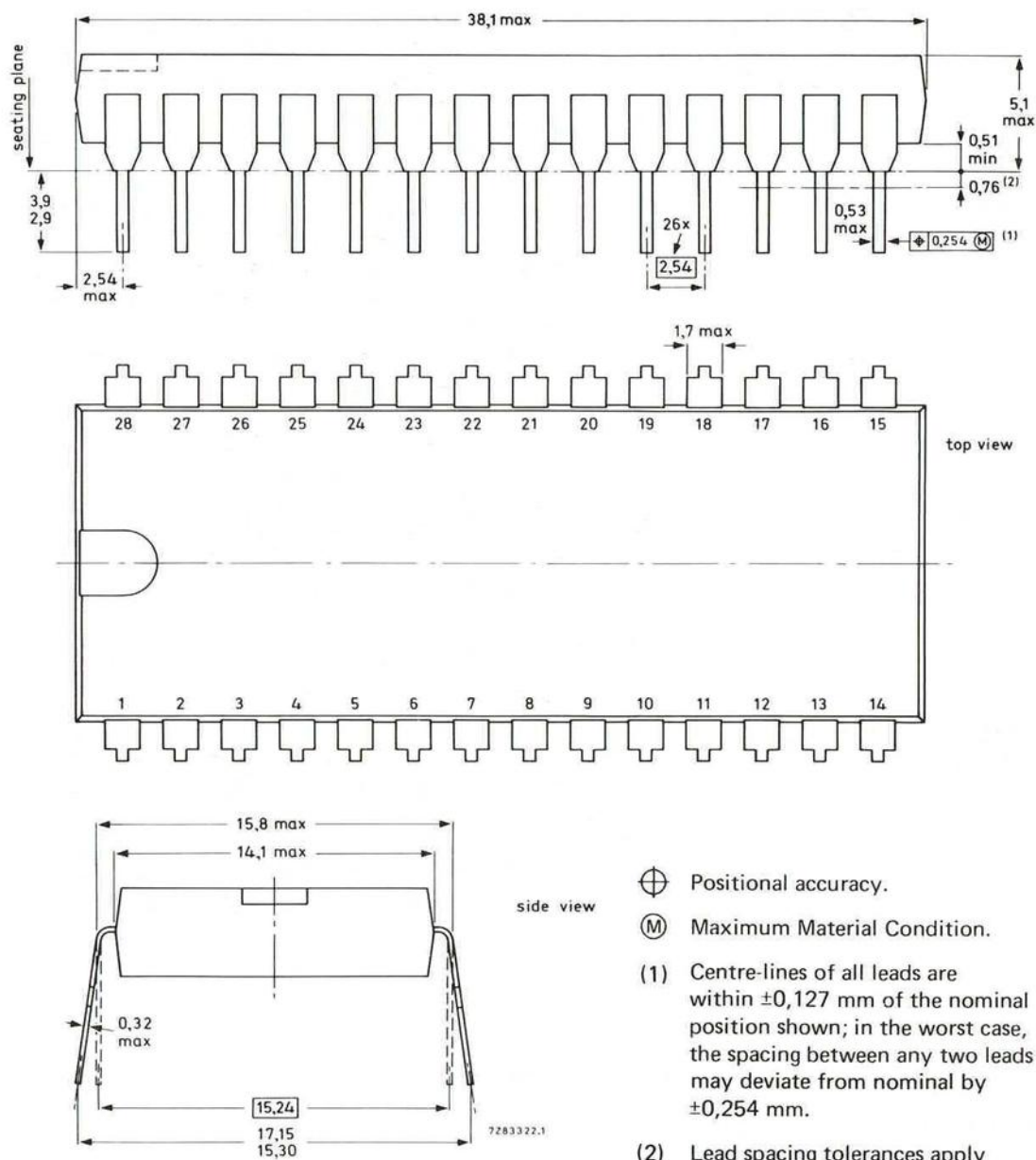
*V_{BB} should be left not connected.



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.

28-LEAD DUAL IN-LINE; PLASTIC (SOT-117D)



⊕ Positional accuracy.

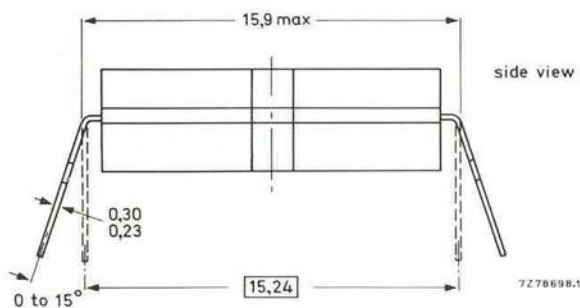
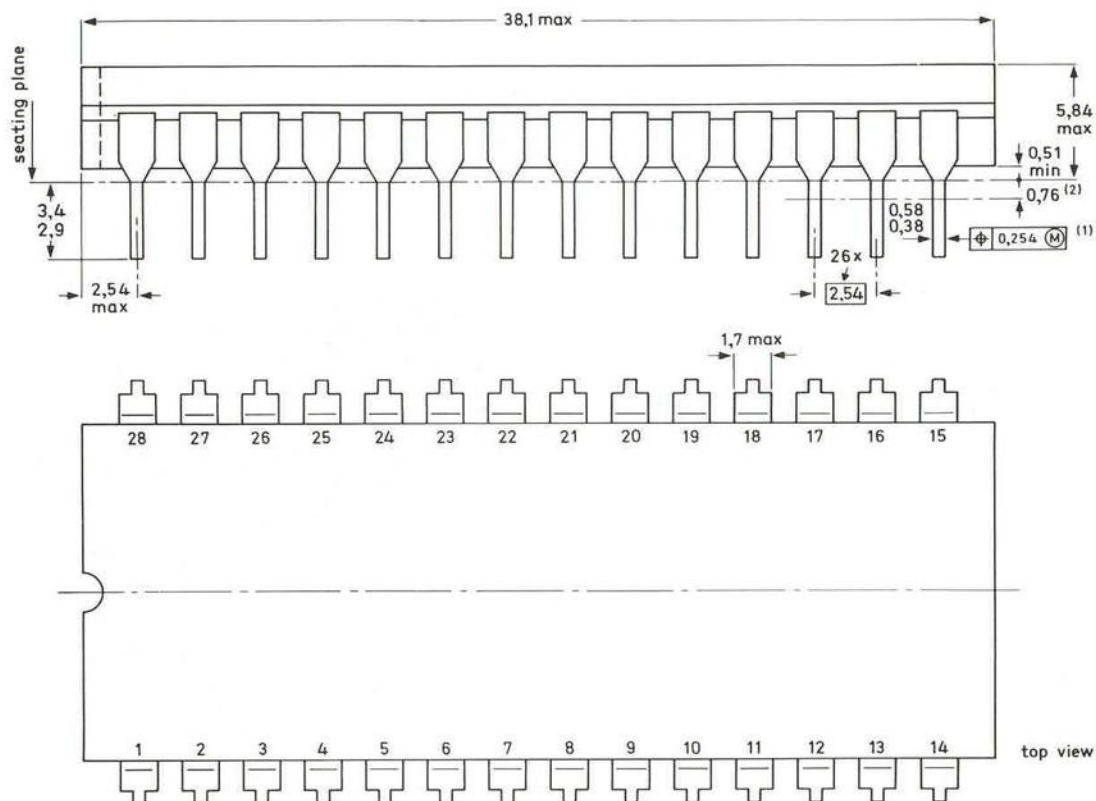
Ⓜ Maximum Material Condition.

(1) Centre-lines of all leads are within $\pm 0,127$ mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by $\pm 0,254$ mm.

(2) Lead spacing tolerances apply from seating plane to the line indicated.

Dimensions in mm

28-LEAD DUAL IN-LINE; CERAMIC (SOT-135)



⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

(1) Centre-lines of all leads are within $\pm 0,127$ mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by $\pm 0,254$ mm.

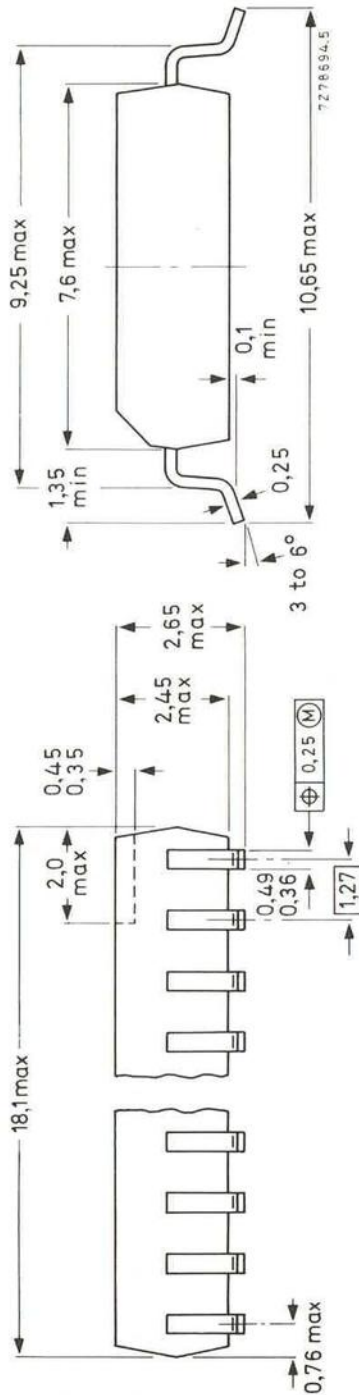
(2) Lead spacing tolerances apply from seating plane to the line indicated.

Dimensions in mm

Remarks

1. Leads are given positive misalignment so that they grip after insertion.
2. Leads are Ni-Fe, pure tin plated.

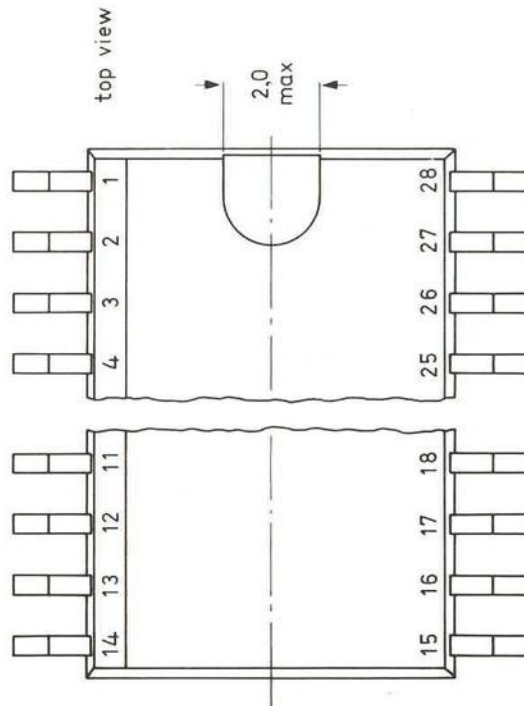
28-LEAD MINI-PACK; PLASTIC (SO-28; SOT-136A)



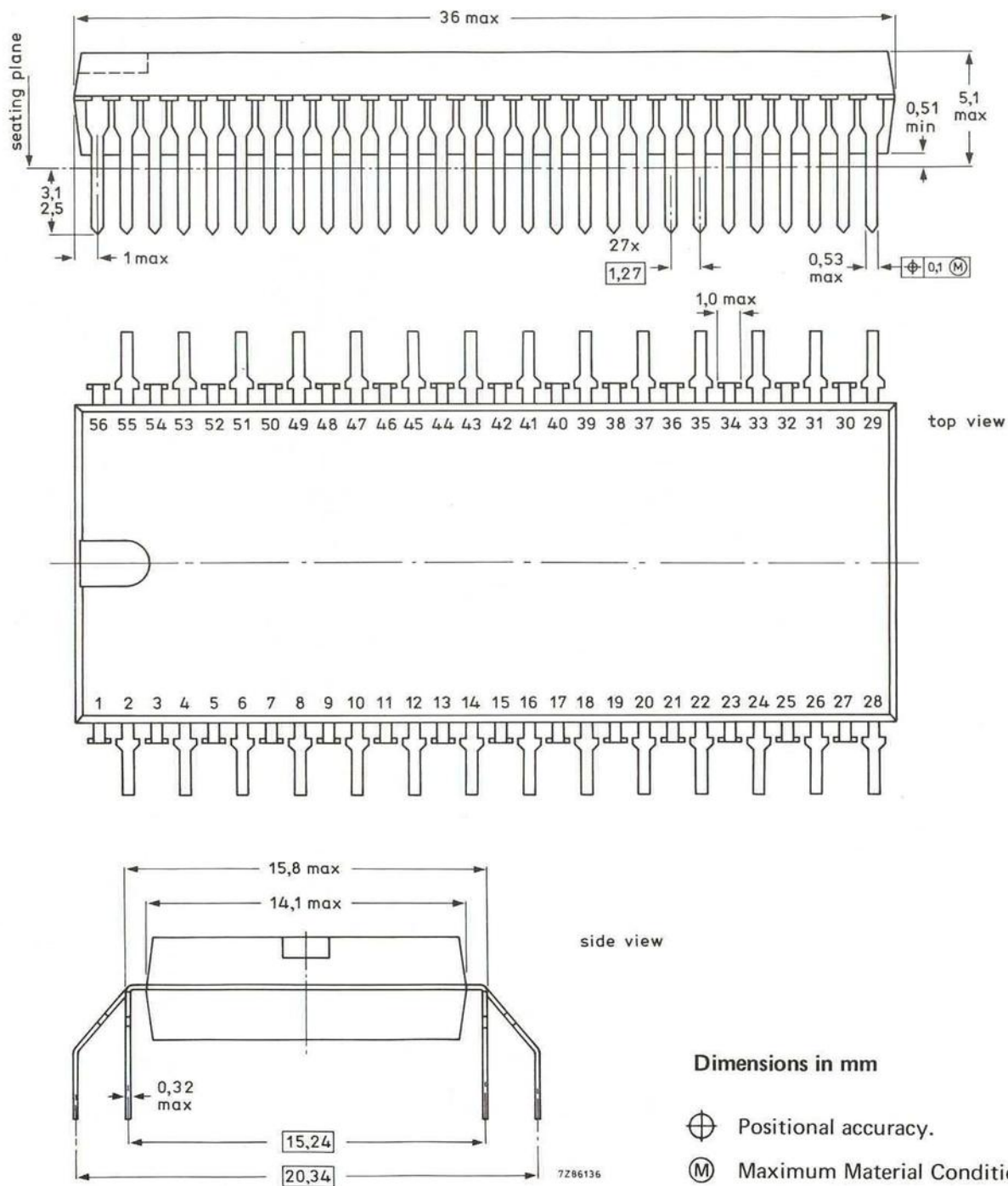
Dimensions in mm

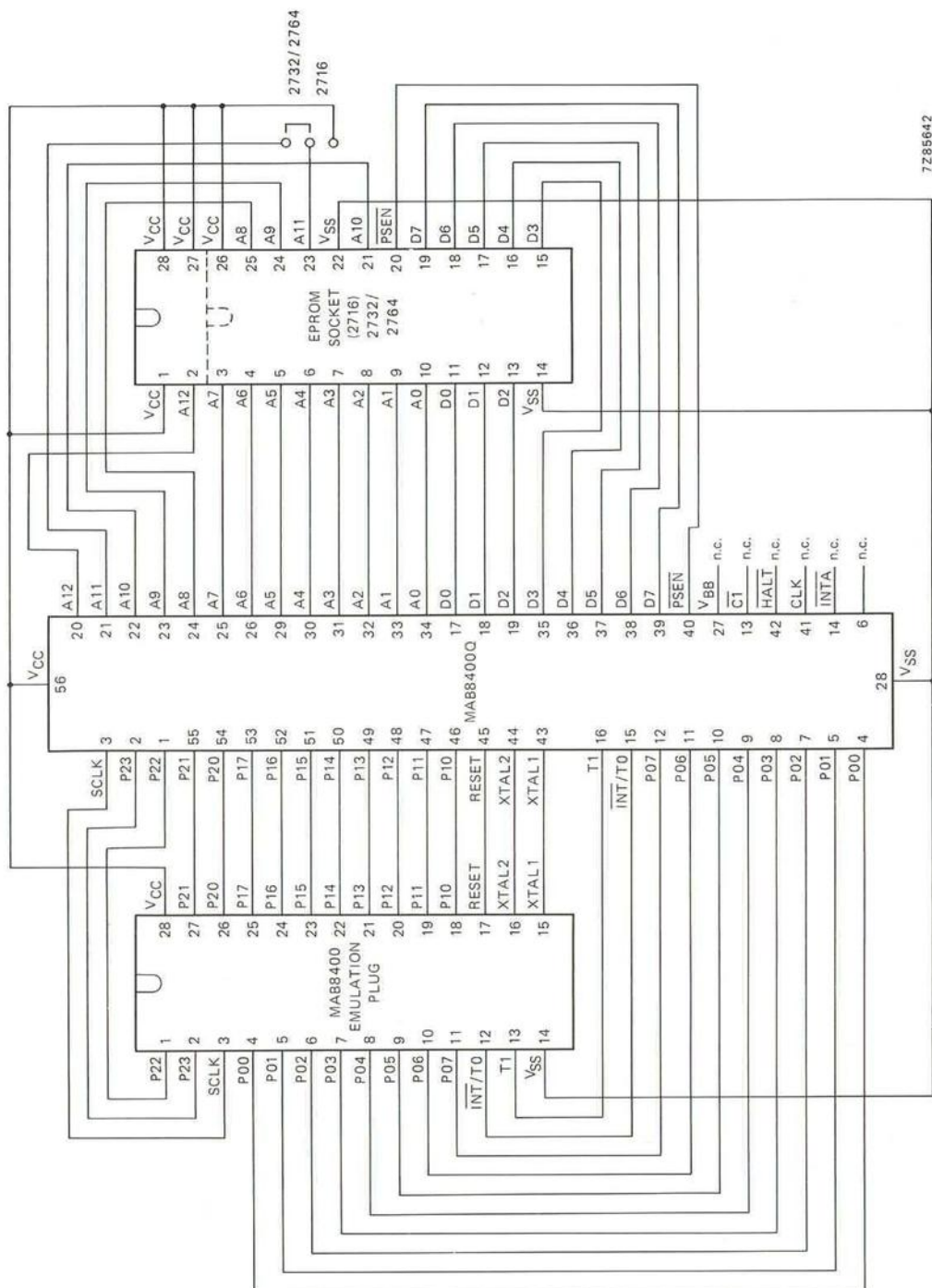
⊕ Positional accuracy.

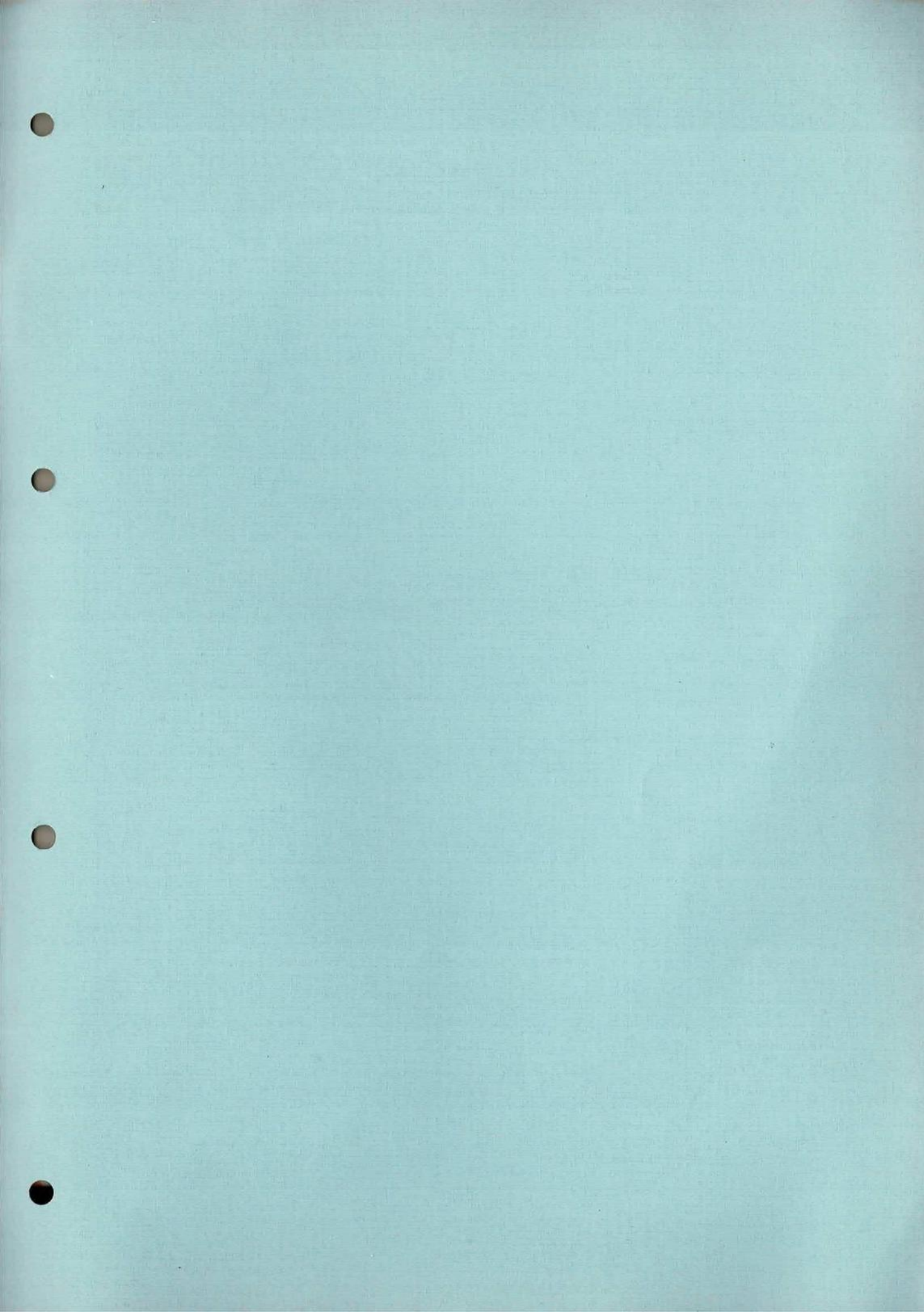
Ⓜ Maximum Material Condition.



56-LEAD QUADRUPLE IN-LINE; PLASTIC (VO-35)







© 1982 N.V. Philips' Gloeilampenfabrieken

This information is furnished for guidance, and with no guarantees as to its accuracy or completeness; its publication conveys no licence under any patent or other right, nor does the publisher assume liability for any consequence of its use; specifications and availability of goods mentioned in it are subject to change without notice; it is not to be reproduced in any way, in whole or in part, without the written consent of the publisher.

Printed in The Netherlands

Date of release: October 1982

9398 614 40011