
MSM80C86A-10RS/GS/JS

16-Bit CMOS MICROPROCESSOR

GENERAL DESCRIPTION

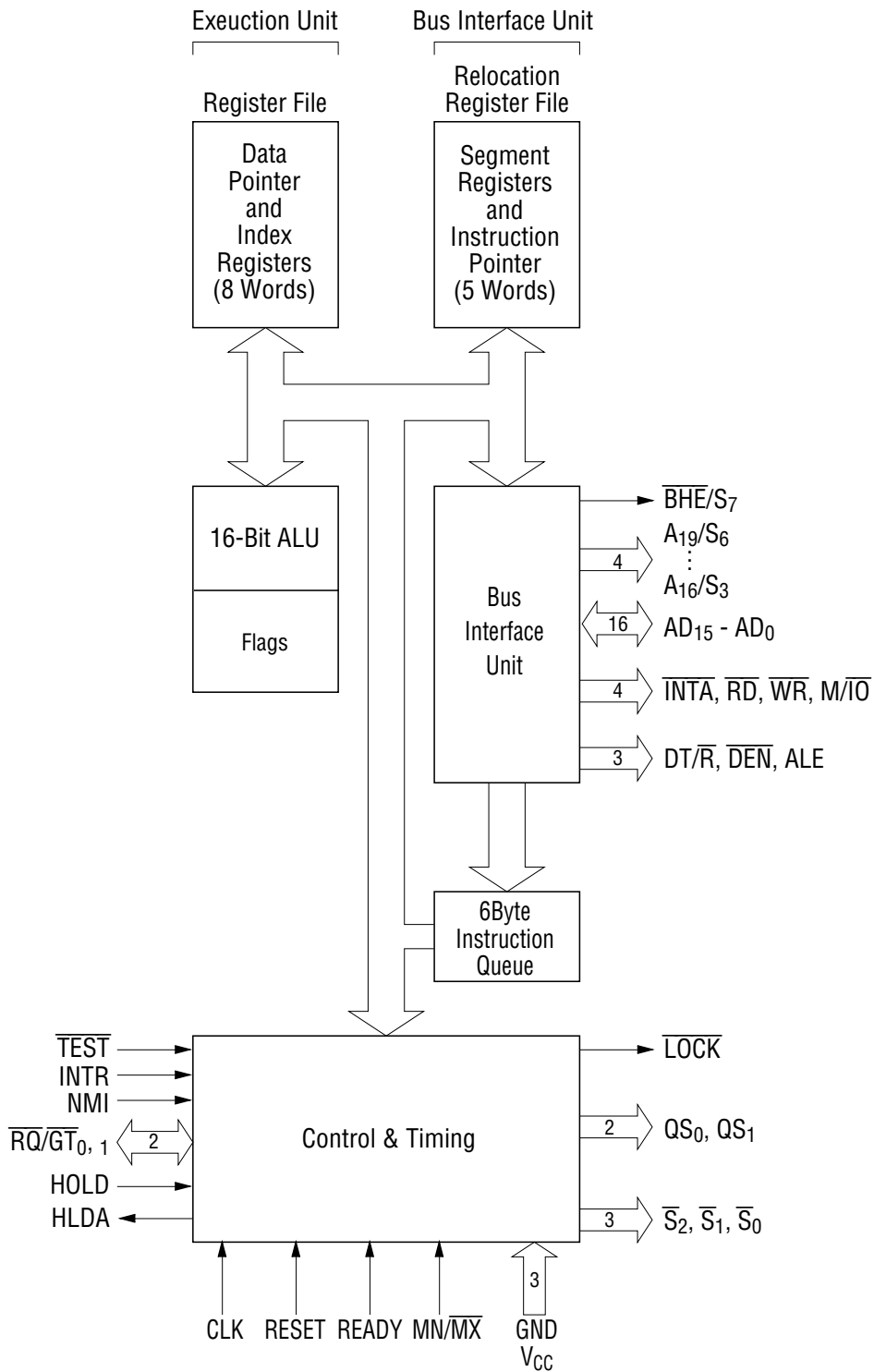
The MSM80C86A-10 is complete 16-bit CPUs implemented in Silicon Gate CMOS technology. They are designed with same processing speed as the NMOS 8086-1 but have considerably less power consumption. It is directly compatible with MSM80C88A-10 software and MSM80C85AH hardware and peripherals.

FEATURES

- 1 Mbyte Direct Addressable Memory Space
- Internal 14-word by 16-bit Register Set
- 24-Operand Addressing Modes
- Bit, Byte, Word and String Operations
- 8 and 16-bit Signed and Unsigned Arithmetic Operation
- From DC to 10 MHz Clock Rate (Note)
- Low Power Dissipation 10 mA/MHz
- Bus Hold Circuitry Eliminated Pull-up Resistors
- 40-pin Plastic DIP (DIP40-P-600-2.54): (Product name: MSM80C86A-10RS)
- 44-pin Plastic QFJ (QFJ44-P-S650-1.27): (Product name: MSM80C86A-10JS)
- 56-pin Plastic QFP (QFP56-P-1519-1.00-K): (Product name: MSM80C86A-10GS-K)

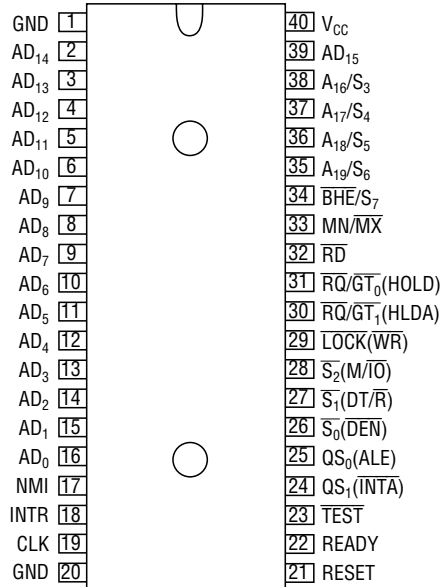
(Note) 10 MHz Spec is not compatible with Intel 8086-1 Spec.

CIRCUIT CONFIGURATION

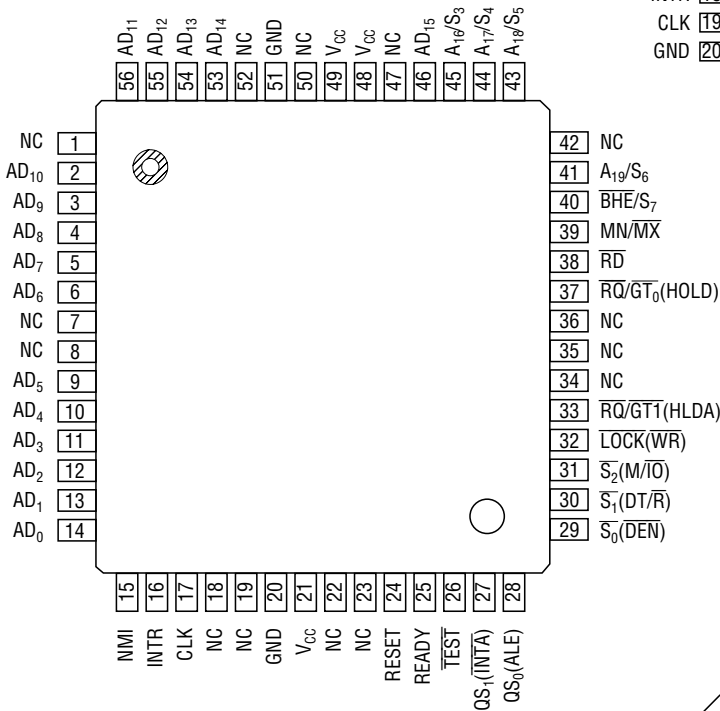


PIN CONFIGURATION (TOP VIEW)

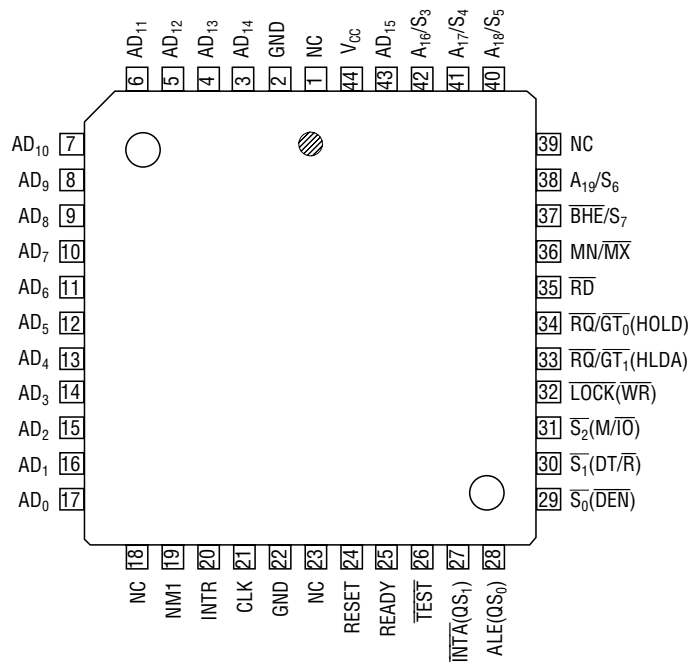
40 pin Plastic DIP



56 pin Plastic QFP



44 pin Plastic QFJ



ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Rating			Unit
			MSM80C86A-10RS	MSM80C86A-10GS	MSM80C86A-10JS	
Power Supply Voltage	V_{CC}	With respect to GND	-0.5 to +7			V
Input Voltage	V_{IN}		-0.5 to $V_{CC}+0.5$			V
Output Voltage	V_{OUT}		-0.5 to $V_{CC}+0.5$			V
Storage Temperature	T_{STG}	—	-65 to +150			°C
Power Dissipation	P_D	$T_a = 25^{\circ}\text{C}$	1.0	0.7		W

OPERATING RANGE

Parameter	Symbol	Range	Unit
Power Supply Voltage	V_{CC}	4.75 to 5.25	V
Operating Temperature	T_{OP}	0 to +70	°C

RECOMMENDED OPERATING CONDITIONS

Parameter	Symbol	Min.	Typ.	Max.	Unit	
Power Supply Voltage	V_{CC}	4.75	5.0	5.25	V	
Operating Temperature	T_{OP}	0	+25	+70	°C	
"L" Input Voltage	V_{IL}	-0.5	—	+0.8	V	
"H" Input Voltage	V_{IH}	*1	$V_{CC}-0.8$	—	$V_{CC}+0.5$	V
		*2	2.0	—	$V_{CC}+0.5$	V

*1 Only CLK

*2 Except CLK

DC CHARACTERISTICS

(V_{CC} = 4.5 to 5.5 V, Ta = -40°C to +85°C)

Parameter	Symbol	Min.	Typ.	Max.	Unit	Conditions
"L" Output Voltage	V _{OL}	—	—	0.4	V	I _{OL} = 2.5 mA
"H" Output Voltage	V _{OH}	3.0	—	—	V	I _{OH} = -2.5 mA
		V _{CC} - 0.4				I _{OH} = -100 μA
Input Leak Current	I _{LI}	-1.0	—	+1.0	μA	0 ≤ V _{IN} ≤ V _{CC}
Output Leak Current	I _{LO}	-10	—	+10	μA	V _O = V _{CC} or GND
Input Leakage Current (Bus Hold Low)	I _{BHL}	50	—	400	μA	V _{IN} = 0.8 V *3
Input Leakage Current (Bus Hold High)	I _{BHH}	-50	—	-400	μA	V _{IN} = 3.0 V *4
Bus Hold Low Overdrive	I _{BHLO}	—	—	600	μA	*5
Bus Hold High Overdrive	I _{BHHO}	—	—	-600	μA	*6
Operating Power Supply Current	I _{CC}	—	—	10	mA/MHz	V _{IL} = GND V _{IH} = V _{CC}
Standby Power Supply Current	I _{CCS}	—	—	500	μA	V _{CC} = 5.5 V Outputs Unloaded V _{IN} = V _{CC} or GND
Input Capacitance	C _{IN}	—	—	10	pF	*7
Output Capacitance	C _{OUT}	—	—	15	pF	*7
I/O Capacitance	C _{I/O}	—	—	20	pF	*7

*3 Test condition is to lower V_{IN} to GND and then raise V_{IN} to 0.8 V on pins 2-16, and 35-39.

*4 Test condition is to raise V_{IN} to V_{CC} and then lower V_{IN} to 3.0 V on pins 2-16, 26-32, and 34-39.

*5 An external driver must source at least I_{BHLO} to switch this node from LOW to HIGH.

*6 An external driver must sink at least I_{BHHO} to switch this node from HIGH to LOW.

*7 Test Conditions: a) Freq = 1 MHz.

b) Unmeasured Pins at GND.

c) V_{IN} at 5.0 V or GND.

AC CHARACTERISTICS

Minimum Mode System
Timing Requirements

Parameter	Symbol	5 MHz Spec. V _{CC} = 4.5 V to 5.5 V Ta = -40 to +85°C		8 MHz Spec. V _{CC} = 4.75 V to 5.25 V Ta = 0 to +70°C		10 MHz Spec. V _{CC} = 4.75 V to 5.25 V Ta = 0 to +70°C		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
CLK Cycle Period	t _{CLCL}	200	DC	125	DC	100	DC	ns
CLK Low Time	t _{CLCH}	118	—	68	—	46	—	ns
CLK High Time	t _{CHCL}	69	—	44	—	44	—	ns
CLK Rise Time (From 1.0 V to 3.5 V)	t _{CH1CH2}	—	10	—	10	—	10	ns
CLK Fall Time (From 3.5 V to 1.0 V)	t _{CL2CL1}	—	10	—	10	—	10	ns
Data in Setup Time	t _{DVCL}	30	—	20	—	20	—	ns
Data in Hold Time	t _{CLDX}	10	—	10	—	10	—	ns
RDY Setup Time into MSM 82C84A-2 (See Notes 1, 2)	t _{R1VCL}	35	—	35	—	35	—	ns
RDY Hold Time into MSM 82C84A-2 (See Notes 1, 2)	t _{CLR1X}	0	—	0	—	0	—	ns
READY Setup Time into MSM80C86A-2	t _{RYHCH}	118	—	68	—	46	—	ns
READY Hold Time into MSM80C86A-10	t _{CHRYX}	30	—	20	—	20	—	ns
READY inactive to CLK (See Note 3)	t _{RYLCL}	-8	—	-8	—	-8	—	ns
HOLD Setup Time	t _{HVCH}	35	—	20	—	20	—	ns
INTR, NMI, $\overline{\text{TEST}}$ Setup Time (See Note 2)	t _{INVCH}	30	—	15	—	15	—	ns
Input Rise Time (Except CLK) (From 0.8 V to 2.0 V)	t _{ILIH}	—	15	—	15	—	15	ns
Input Fall Time (Except CLK) (From 2.0 V to 0.8 V)	t _{IHIL}	—	15	—	15	—	15	ns

Timing Responses

Parameter	Symbol	5 MHz Spec. V _{CC} = 4.5 V to 5.5 V Ta = -40 to +85°C		8 MHz Spec. V _{CC} = 4.75 V to 5.25 V Ta = 0 to +70°C		10 MHz Spec. V _{CC} = 4.75 V to 5.25 V Ta = 0 to +70°C		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
		Address Valid Delay	t _{CLAV}	10	110	10	60	
Address Hold Time	t _{CLAX}	10	—	10	—	10	—	ns
Address Float Delay	t _{CLAZ}	t _{CLAX}	80	t _{CLAX}	50	t _{CLAX}	50	ns
ALE Width	t _{LHLL}	t _{CLCH-20}	—	t _{CLCH-10}	—	t _{CLCH-10}	—	ns
ALE Active Delay	t _{CLLH}	—	80	—	50	—	40	ns
ALE Inactive Delay	t _{CHLL}	—	85	—	55	—	45	ns
Address Hold Time to ALE Inactive	t _{LLAX}	t _{CLCH-10}	—	t _{CLCH-10}	—	t _{CLCH-10}	—	ns
Data Valid Delay	t _{CLDV}	10	110	10	60	10	60	ns
Data Hold Time	t _{CHDX}	10	—	10	—	10	—	ns
Data Hold Time after \overline{WR}	t _{WHDX}	t _{CLCH-30}	—	t _{CLCH-30}	—	t _{CLCH-25}	—	ns
Control Active Delay 1	t _{CVCTV}	10	110	10	70	10	55	ns
Control Active Delay 2	t _{CHCTV}	10	110	10	60	10	50	ns
Control Inactive Delay	t _{CVCTX}	10	110	10	70	10	55	ns
Address Float to \overline{RD} Active	t _{AZRL}	0	—	0	—	0	—	ns
\overline{RD} Active Delay	t _{CLRL}	10	165	10	100	10	70	ns
\overline{RD} Inactive Delay	t _{CLRH}	10	150	10	80	10	60	ns
\overline{RD} Inactive to Next Address Active	t _{RHAV}	t _{CLC-45}	—	t _{CLCH-40}	—	t _{CLCL-35}	—	ns
HLDA Valid Delay	t _{CLHAV}	10	160	10	100	10	60	ns
\overline{RD} Width	t _{RLRH}	2t _{CLCL-75}	—	2t _{CLCL-50}	—	2t _{CLCL-40}	—	ns
\overline{WR} Width	t _{WLWH}	2t _{CLCL-60}	—	2t _{CLCL-40}	—	2t _{CLCL-35}	—	ns
Address Valid to ALE Low	t _{AVAL}	t _{CLCH-60}	—	t _{CLCH-40}	—	t _{CLCH-35}	—	ns
Output Rise Time (From 0.8 V to 2.0 V)	t _{OLOH}	—	15	—	15	—	15	ns
Output Fall Time (From 2.0 V to 0.8 V)	t _{OHOL}	—	15	—	15	—	15	ns

- Notes:
1. Signal at MSM82C84A-2 or MSM82C88-2 are shown for reference only.
 2. Setup requirement for asynchronous signal only to guarantee recognition at next CLK.
 3. Applies only to T2 state. (8 ns into T3)

Maximum Mode System (Using MSM82C88-2 Bus Controller)

Timing Requirements

Parameter	Symbol	5 MHz Spec. V _{CC} = 4.5 V to 5.5 V Ta = -40 to +85°C		8 MHz Spec. V _{CC} = 4.75 V to 5.25 V Ta = 0 to +70°C		10 MHz Spec. V _{CC} = 4.75 V to 5.25 V Ta = 0 to +70°C		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
		CLK Cycle Period	t _{CLCL}	200	DC	125	DC	
CLK Low Time	t _{CLCH}	118	—	68	—	46	—	ns
CLK High Time	t _{CHCL}	69	—	44	—	44	—	ns
CLK Rise Time (From 1.0 V to 3.5 V)	t _{CH1CH2}	—	10	—	10	—	10	ns
CLK Fall Time (From 3.5 V to 1.0 V)	t _{CL2CL1}	—	10	—	10	—	10	ns
Data in Setup Time	t _{DVCL}	30	—	20	—	20	—	ns
Data in Hold Time	t _{CLDX}	10	—	10	—	10	—	ns
RDY Setup Time into MSM 82C84A-2 (See Notes 1, 2)	t _{R1VCL}	35	—	35	—	35	—	ns
RDY Hold Time into MSM 82C84A-2 (See Notes 1, 2)	t _{CLR1X}	0	—	0	—	0	—	ns
READY Setup Time into MSM80C86A-10	t _{RYHCH}	118	—	68	—	46	—	ns
READY Hold Time into MSM80C86A-10	t _{CHRYX}	30	—	20	—	20	—	ns
READY inactive to CLK (See Note 3)	t _{RYLCL}	-8	—	-8	—	-8	—	ns
Setup Time for Recognition (NMI, INTR, $\overline{\text{TEST}}$) (See Note 2)	t _{INVCH}	30	—	15	—	15	—	ns
$\overline{\text{RQ}}/\overline{\text{GT}}$ Setup Time	t _{GVCH}	30	—	15	—	15	—	ns
$\overline{\text{RQ}}$ Hold Time into MSM80C86A-10	t _{CHGX}	40	—	30	—	20	—	ns
Input Rise Time (Except CLK) (From 0.8 V to 2.0 V)	t _{LIH}	—	15	—	15	—	15	ns
Input Fall Time (Except CLK) (From 2.0 V to 0.8 V)	t _{HIL}	—	15	—	15	—	15	ns

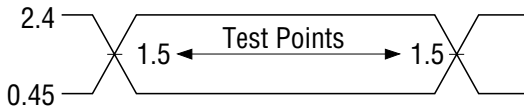
Timing Responses

Timing Response Parameter	Symbol	5 MHz Spec. V _{CC} = 4.5 V to 5.5 V T _a = -40 to +85°C		8 MHz Spec. V _{CC} = 4.75 V to 5.25 V T _a = 0 to +70°C		10 MHz Spec. V _{CC} = 4.75 V to 5.25 V T _a = 0 to +70°C		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
Command Active Delay (See Note 1)	t _{CLML}	5	45	5	35	5	35	ns
Command Inactive Delay (See Note 1)	t _{CLMH}	5	45	5	45	5	45	ns
READY Active to Status Passive (See Note 4)	t _{RYHSH}	—	110	—	65	—	45	ns
Status Active Delay	t _{CHSV}	10	110	10	60	10	45	ns
Status Inactive Delay	t _{CLSH}	10	130	10	70	10	60	ns
Address Valid Delay	t _{CLAV}	10	110	10	60	10	60	ns
Address Hold Time	t _{CLAX}	10	—	10	—	10	—	ns
Address Float Delay	t _{CLAZ}	t _{CLAX}	80	t _{CLAX}	50	t _{CLAX}	50	ns
Status Valid to ALE High (See Note 1)	t _{SVLH}	—	35	—	25	—	25	ns
Status Valid to MCE High (See Note 1)	t _{SVMCH}	—	35	—	30	—	30	ns
CLK Low to ALE Valid (See Note 1)	t _{CLLH}	—	35	—	25	—	25	ns
CLK Low to MCE High (See Note 1)	t _{CLMCH}	—	35	—	25	—	25	ns
ALE Inactive Delay (See Note 1)	t _{CHLL}	4	35	4	25	4	25	ns
Data Valid Delay	t _{CLDV}	10	110	10	60	10	60	ns
Data Hold Time	t _{CHDX}	10	—	10	—	10	—	ns
Control Active Delay (See Note 1)	t _{CVNV}	5	45	5	45	5	45	ns
Control Inactive Delay (See Note 1)	t _{CVNX}	5	45	5	45	5	45	ns
Address Float to RD Active	t _{AZRL}	0	—	0	—	0	—	ns
RD Active Delay	t _{CLRL}	10	165	10	100	10	70	ns
RD Inactive Delay	t _{CLRH}	10	150	10	80	10	60	ns
RD Inactive to Next Address Active	t _{RHAV}	t _{CLCL-45}	—	t _{CLCL-40}	—	t _{CLCL-35}	—	ns
Direction Control Active Delay (See Note 1)	t _{CHDTL}	—	50	—	50	—	50	ns
Direction Control Inactive Delay (See Note 1)	t _{CHDTH}	—	35	—	30	—	30	ns
GT Active Delay (See Note 5)	t _{CLGL}	0	85	0	50	0	45	ns
GT Inactive Delay	t _{CLGH}	0	85	0	50	0	45	ns
RD Width	t _{RLRH}	2t _{CLCL-75}	—	2t _{CLCL-50}	—	2t _{CLCL-40}	—	ns
Output Rise Time (From 0.8 V to 2.0 V)	t _{OLOH}	—	15	—	15	—	15	ns
Output Fall Time (From 2.0 V to 0.8 V)	t _{OHOL}	—	15	—	15	—	15	ns

- Notes: 1. Signals at MSM82C84A-2 or MSM82C88-2 are shown for reference only.
 2. Setup requirement for asynchronous signal only to guarantee recognition at next CLK
 3. Applies only to T2 state (8 ns into T3)
 4. Applies only to T3 and wait states.
 5. C_L = 40 pF (RQ/GT₀, RQ/GT₁)

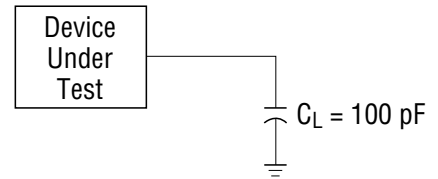
TIMING DIAGRAM

Input/Output



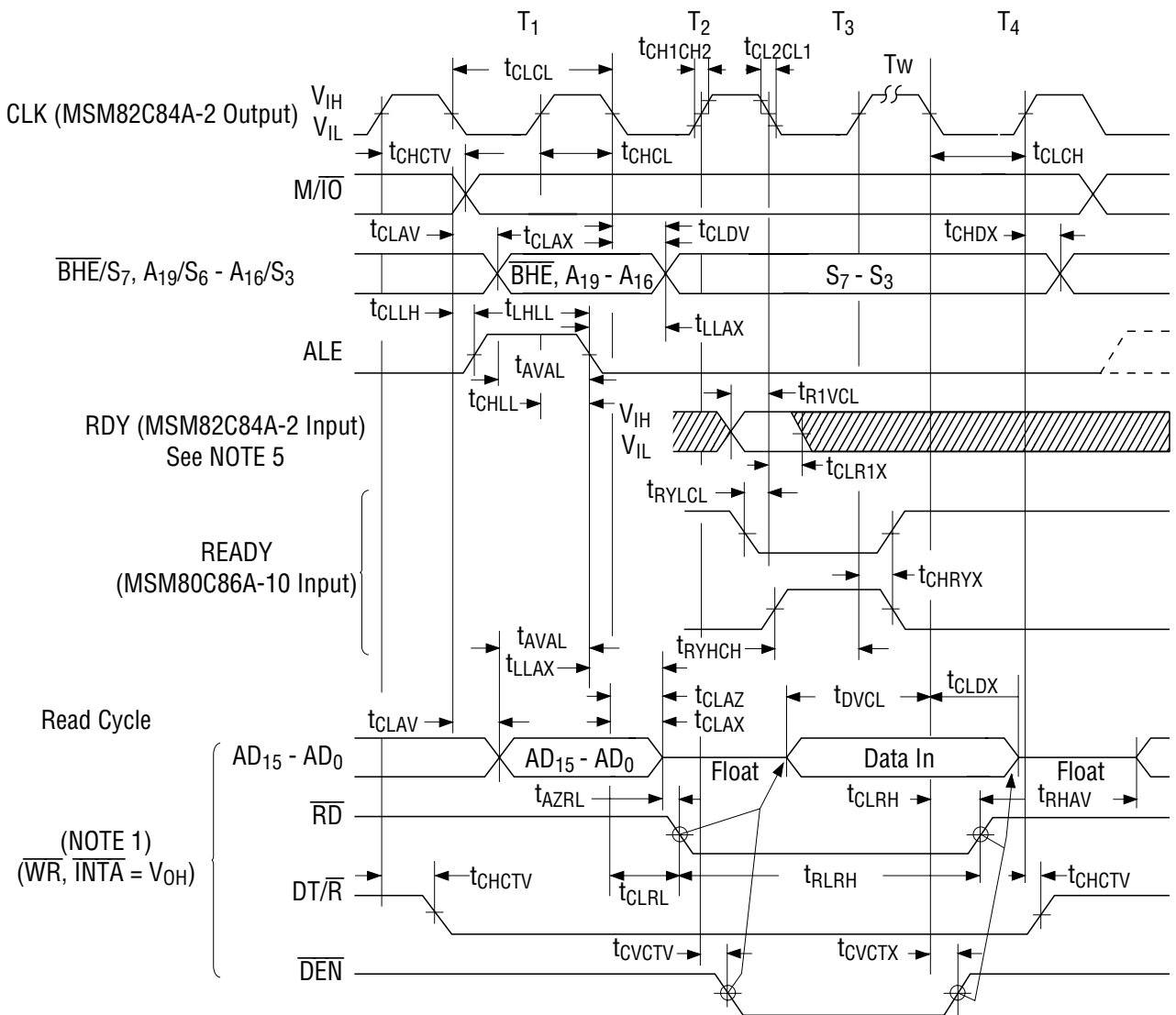
AC, Testing: Inputs are driven at 2.4 V for a logic "1" and 0.45 V for a logic "0". Timing measurements are 1.5 V for both a logic "1" and "0".

A.C. Testing Load Circuit



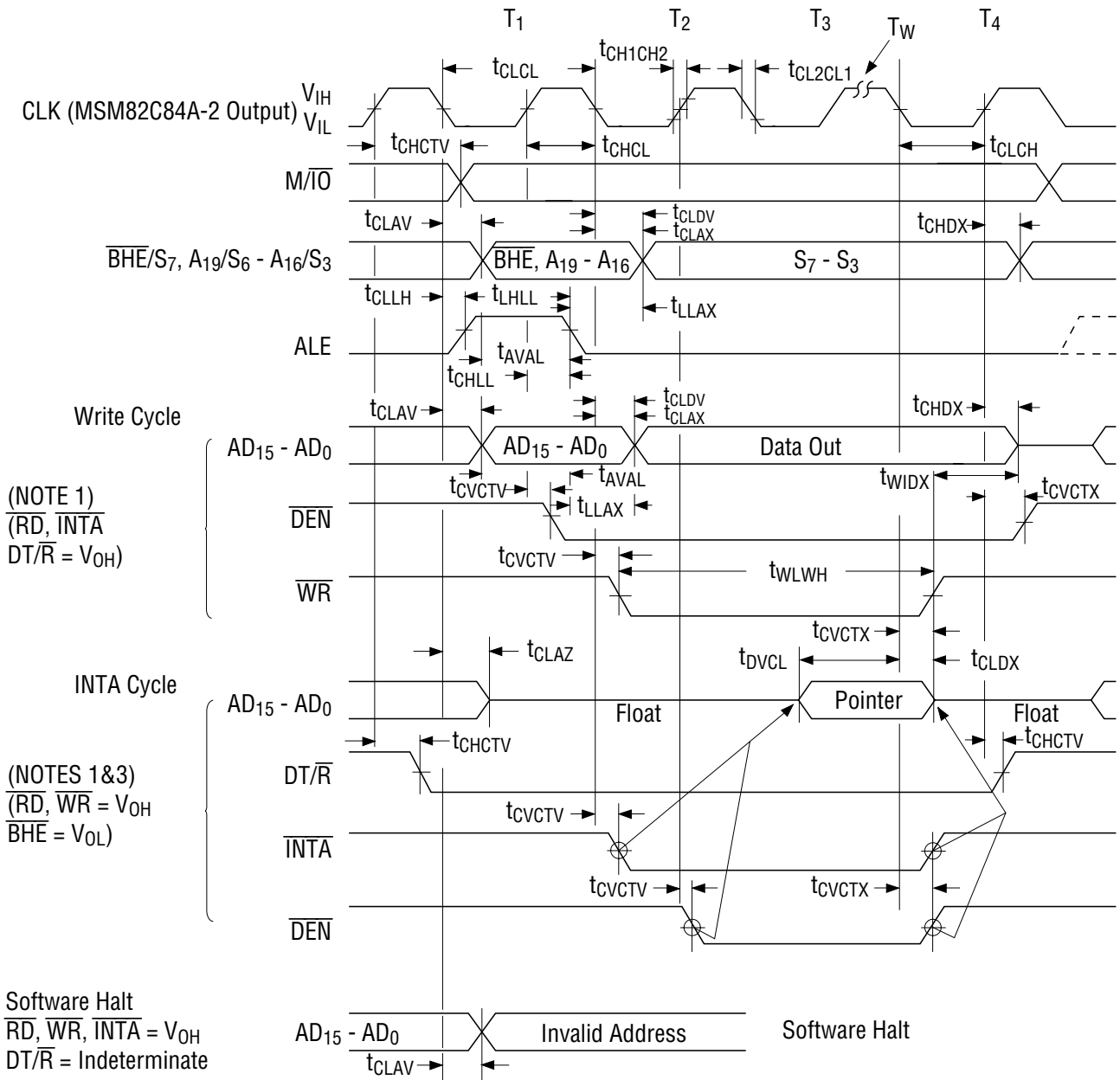
C_L includes jig capacitance.

Minimum Mode

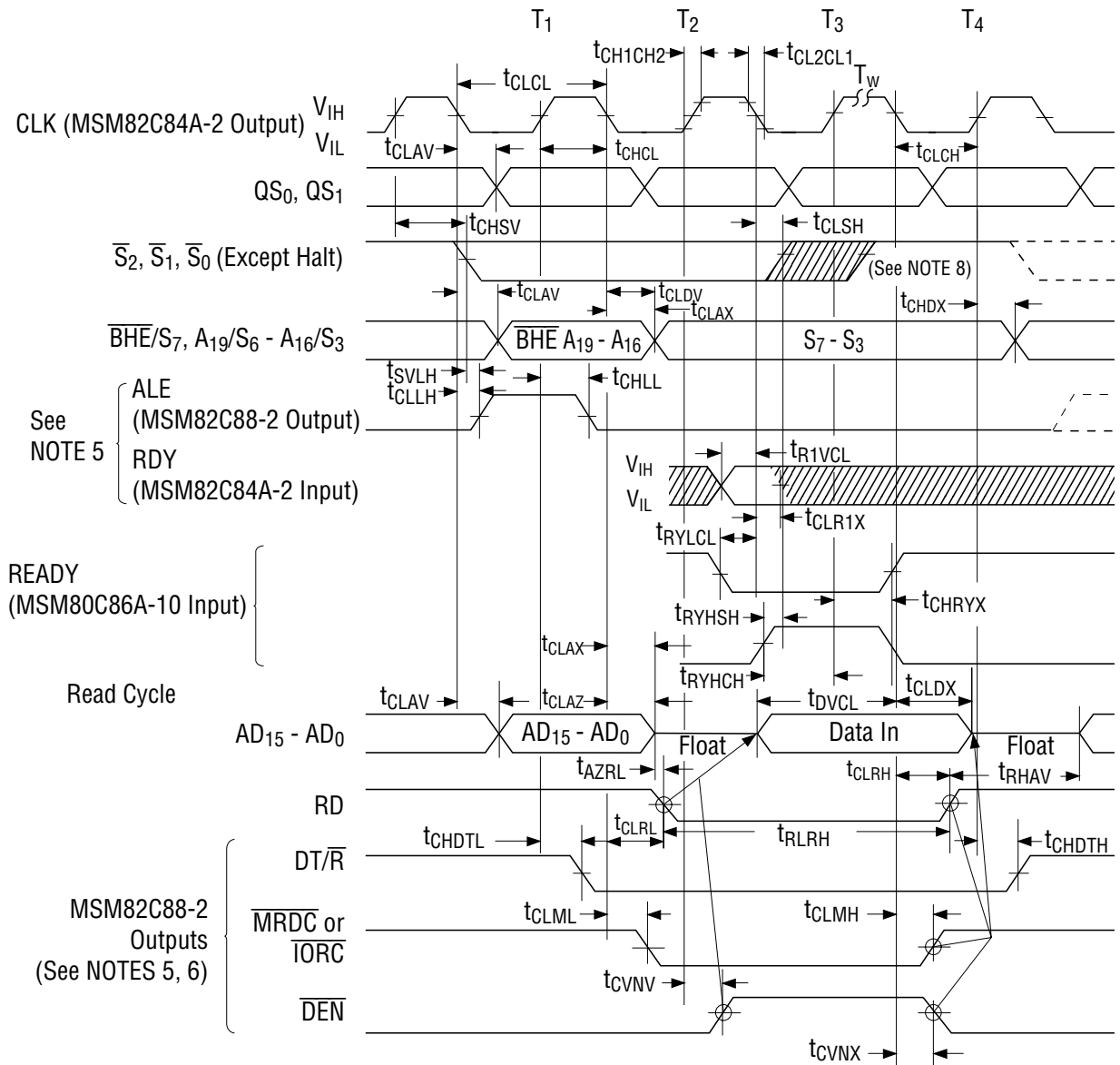


(NOTE 1)
(WR, INTA = V_{OH})

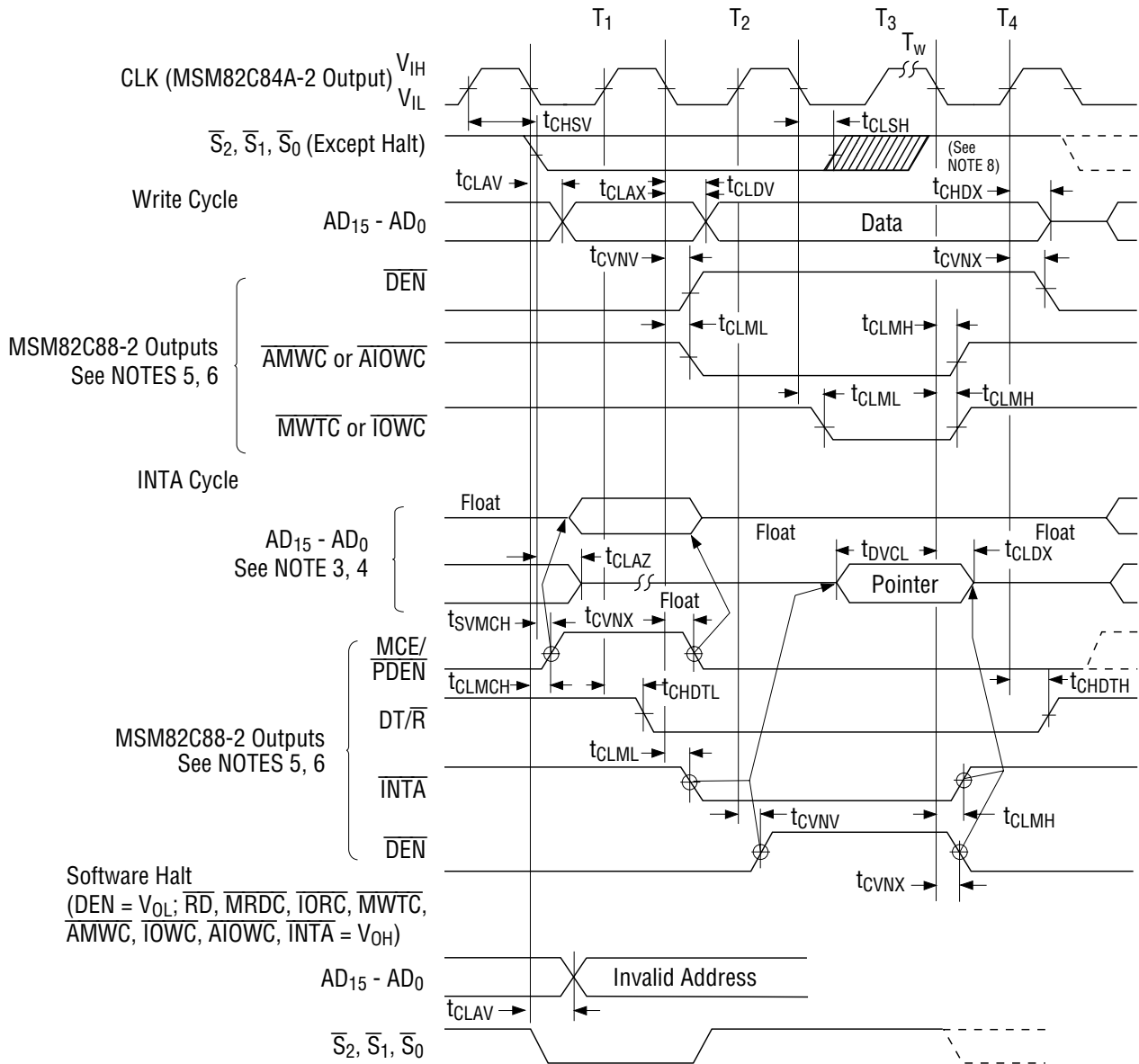
Minimum Mode (continued)



Maximum Mode

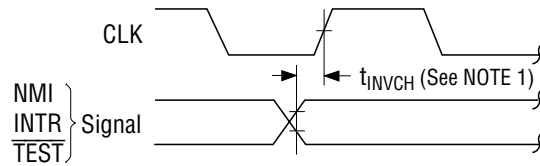


Maximum Mode (continued)



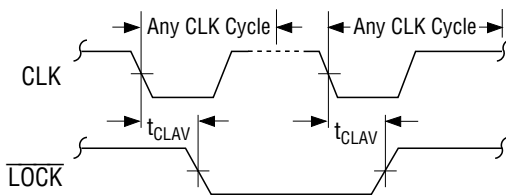
- Notes:
1. All signals switch between V_{OH} and V_{OL} unless otherwise specified.
 2. RDY is sampled near the end of T₂, T₃, T_W to determine if T_W machines states are to be inserted.
 3. Cascade address is valid between first and second INTA cycle.
 4. Two INTA cycles run back-to-back. The MSM80C86A-10 LOCAL ADDR/DATA BUS is floating during both INTA cycles. Control for pointer address is shown for second INTA cycle.
 5. Signals at MSM82C84A-2 or MSM82C88-2 are shown for reference only.
 6. The issuance of the MSM 82C88-2 command and control signals (MRDC, MWTC, AMWC, IORC, IOWC, AIOWC, INTA and DEN) lags the active high MSM82C88-2 CEN.
 7. All timing measurements are made at 1.5 V unless otherwise noted.
 8. Status inactive in state just prior to T₄

Asynchronous Signal Recognition

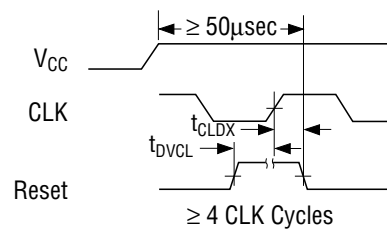


NOTE: 1 Setup requirements for asynchronous signals only to guarantee recognition at next CLK

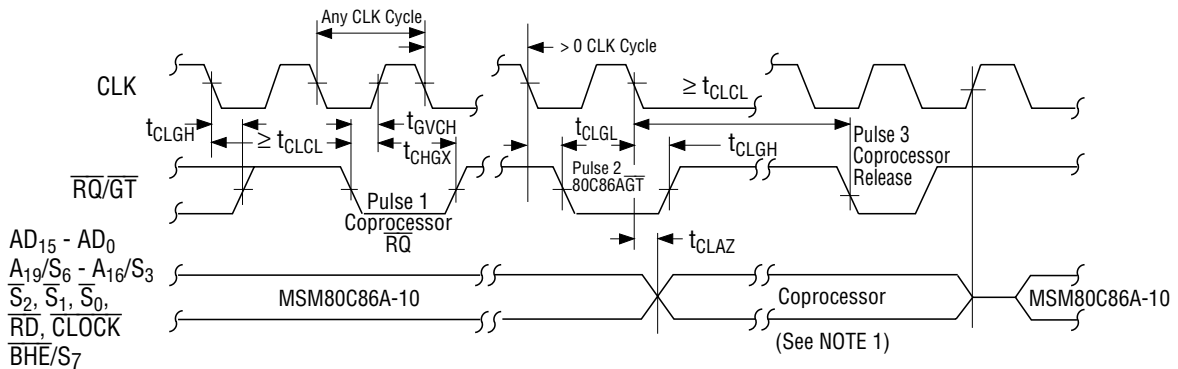
Bus Lock Signal Timing (Maximum Mode Only)



Reset Timing

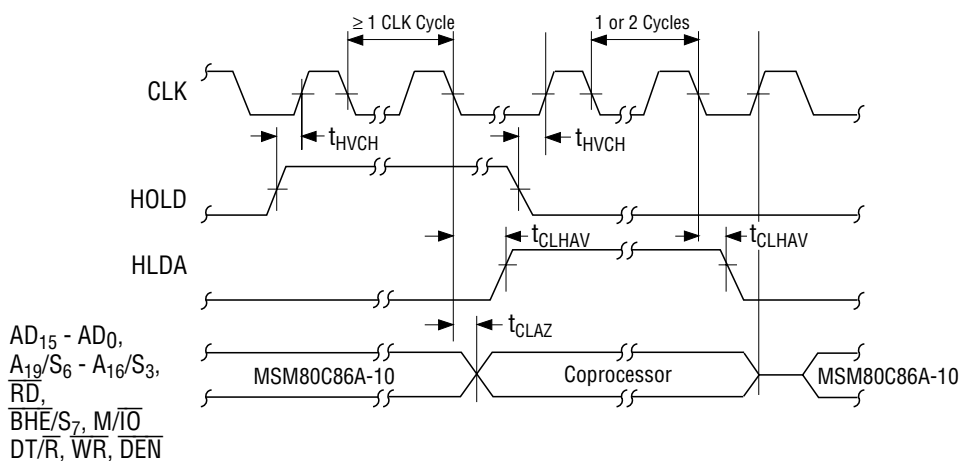


Request/Grant Sequence Timing (Maximum Mode Only)



NOTE: 1 The coprocessor may not drive the buses outside the region shown without risking contention.

Hold/Hold Acknowledge Timing (Minimum Mode Only)



PIN DESCRIPTION

AD₀ - AD₁₅

ADDRESS DATA BUS: Input/Output

These lines are the multiplexed address and data bus.

These are the address bus at the T1 cycle and the data bus at the T2, T3, TW and T4 cycles.

At the T1 cycle, AD₀ low indicates Data Bus Low (D₀-D₇) Enable. These lines are high impedance during interrupt acknowledge and hold acknowledge.

A₁₆/S₃, A₁₇/S₄, A₁₈/S₅, A₁₉/S₆

ADDRESS/STATUS: Output

These are the four most significant addresses, at the T1 cycle. Accessing I/O port address, these are low at T1 cycles. These lines are Status lines at T2, T3, TW and T4 cycles. S₃ and S₄ are encoded as shown.

S ₃	S ₄	Characteristics
0	0	Alternate Data
1	0	Stack
0	1	Code or None
1	1	Data

These lines are high impedance during hold acknowledge.

$\overline{\text{BHE}}$ /S₇

BUS HIGH ENABLE/STATUS: Output

This line indicates Data Bus High Enable (BHE) at the T1 cycle. This line is status line at T2, T3, TW and T4 cycles.

$\overline{\text{RD}}$

READ: Output

This line indicates that CPU is in the memory or I/O read cycle.

This line is the read strobe signal when CPU read data from memory or I/O device. This line is active low.

This line is high impedance during hold acknowledge.

READY

READY:Input

This line indicates to the CPU that the addressed memory or I/O device is ready to read or write.

This line is active high. If the setup and hold time is out of specification, illegal operation will occur.

INTR

INTERRUPT REQUEST: Input

This line is the level triggered interrupt request signal which is sampled during the last clock cycle of instruction and string manipulation.

It can be internally masked by software.

This signal is active high and internally synchronized.

$\overline{\text{INTA}}$

INTERRUPT ACKNOWLEDGE: Output

This line is a read strobe signal for the interrupt acknowledge cycle. This line is active low.

 $\overline{\text{TEST}}$

TEST: Input

This line is examined by the WAIT instruction.

When $\overline{\text{TEST}}$ is high, the CPU enters idle cycle.

When $\overline{\text{TEST}}$ is low, the CPU exits the idle cycle.

NMI

NON MASKABLE INTERRUPT: Input

This line causes a type 2 interrupt.

NMI is not maskable.

This signal is internally synchronized and needs 2-clock cycles of pulse width.

RESET

RESET: Input

This signal causes the CPU to initialize immediately.

This signal is active high and must be at least four clock cycles.

CLK

CLOCK: Input

This signal provides the basic timing for the internal circuit.

 $\text{MN}/\overline{\text{MX}}$

MINIMUM/MAXIMUM: Input

This signal selects the CPU's operating mode.

When V_{CC} is connected, the CPU operates in Minimum mode.

When GND is connected, the CPU operates in Maximum mode.

 V_{CC}

V_{CC} : +5V supplied.

GND

GROUND

The following pin function descriptions are maximum mode only. Other pin functions are already described.

 S_0, S_1, S_2

STATUS: Output

These lines indicate bus status and they are used by the MSM82C88-2 Bus Controller to generate all memory and I/O access control signals.

These lines are high impedance during hold acknowledge. These status lines are encoded as shown.

$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	Characteristics
0 (LOW)	0	0	Interrupt acknowledge
0	0	1	Read I/O Port
0	1	0	Write I/O Port
0	1	1	Halt
1 (HIGH)	0	0	Code Access
1	0	1	Read Memory
1	1	0	Write Memory
1	1	1	Passive

 $\overline{RQ/GT_0}$ **$\overline{RQ/GT_1}$**

REQUEST/GRANT:Input/Output

These lines are used for Bus Request from other devices and Bus GRANT to other devices. These lines are bidirectional and active low.

 \overline{LOCK}

LOCK:Output

This line is active low.

When this line is low, other devices cannot gain control of the bus.

This line is high impedance during hold acknowledge.

 QS_0/QS_1

QUEUE STATUS: Output

These lines are Queue Status, and indicate internal instruction queue status.

QS_1	QS_0	Characteristics
0 (LOW)	0	No operation
0	1	First Byte of Op Code from Queue
1 (HIGH)	0	Empty the Queue
1	1	Subsequent Byte from Queue

The following pin function descriptions are minimum mode only. Other pin functions are already described.

 M/\overline{IO}

STATUS: Output

This line selects memory address space or I/O address space.

When this line is high, the CPU selects memory address space and when it is low, the CPU selects I/O address space.

This line is high impedance during hold acknowledge.

 \overline{WR}

WRITE: Output

This line indicates that the CPU is in the memory or I/O write cycle.

This line is a write strobe signal when the CPU writes data to memory of I/O device.

This line is active low.

This line is high impedance during hold acknowledge.

$\overline{\text{INTA}}$

INTERRUPT ACKNOWLEDGE: Output

This line is a read strobe signal for the interrupt acknowledge cycle. This line is active low.

ALE

ADDRESS LATCH ENABLE: Output

This line is used for latching the address into the MSM82C12 address latch. It is a positive pulse and its trailing edge is used to strobe the address. This line is never floated.

 $\text{DT}/\overline{\text{R}}$

DATA TRANSMIT/RECEIVE: Output

This line is used to control the output enable of the bus transceiver.

When this line is high, the CPU transmits data, and when it is low, the CPU receives data.

This line is high impedance during hold acknowledge.

 $\overline{\text{DEN}}$

DATA ENABLE: Output

This line is used to control the output enable of the bus transceiver.

This line is active low. This line is high impedance during hold acknowledge.

HOLD

HOLD REQUEST: Input

This line is used for Bus Request from other devices.

This line is active high.

HLDA

HOLD ACKNOWLEDGE: Output

This line is used for Bus Grant other devices.

This line is active high.

FUNCTIONAL DESCRIPTION STATIC OPERATION

The MSM80C86A-10 circuitry is of static design. Internal registers, counters and latches are static and require no refresh as with dynamic circuit design. This eliminates the minimum operating frequency restriction placed on other microprocessors. The MSM80C86A-10 can operate from DC to the appropriate upper frequency limit. The processor clock may be stopped in either state (high/low) and held there indefinitely. This type of operation is especially useful for system debug or power critical applications.

The MSM80C86A-10 can be single stepped using only the CPU clock. This state can be maintained as long as is necessary. Single step clock operation allows simple interface circuitry to provide critical information for bringing up your system.

Static design also allows very low frequency operation (down to DC). In a power critical situation, this can provide extremely low power operation since MSM80C86A-10 power dissipation is directly related to operating frequency. As the system frequency is reduced, so is the operating power until, ultimately, at a DC input frequency, MSM80C86A-10 power requirement is the standby current (500 μ A maximum).

General Operation

The internal function of the MSM80C86A-10 consists of a Bus Interface Unit (BIU) and an Execution Unit (EU). These units operate mutually but perform as separate processors.

BIU performs instruction fetch and queueing, operand fetch, DATA read and write address relocation and basic bus control. Instruction pre-fetch is performed while waiting for decoding and execution of instructions. Thus, the CPU's performance is increased. Up to 6-bytes of instructions stream can be queued.

The EU receives pre-fetched instructions from the BIU queue, decodes and executes the instructions, and provides the un-relocated operand address to BIU.

Memory Organization

The MSM80C86A-10 has a 20-bit address to memory. Each address has an 8-bit data width. Memory is organized 00000H to FFFFFH and is logically divided into four segments: code, data, extra data and stack segment. Each segment contains up to 64 Kbytes and locates on a 16-byte boundary. (Fig. 3a)

All memory references are made relative to the segment register which functions in accordance with a select rule. Word operands can be located on even or odd address boundary.

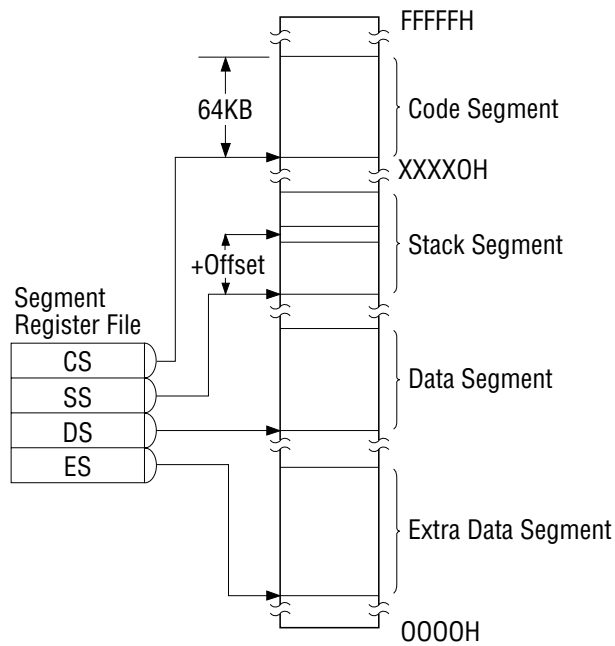
The BIU automatically performs the proper number of memory accesses. Memory consists of an even address and an odd address. Byte data of even address is transferred on the AD₀-AD₇ and byte data of odd address is transferred on the AD₈-AD₁₅.

The CPU provides two enable signals $\overline{\text{BHE}}$ and A₀ to access either an odd address, even address or both:

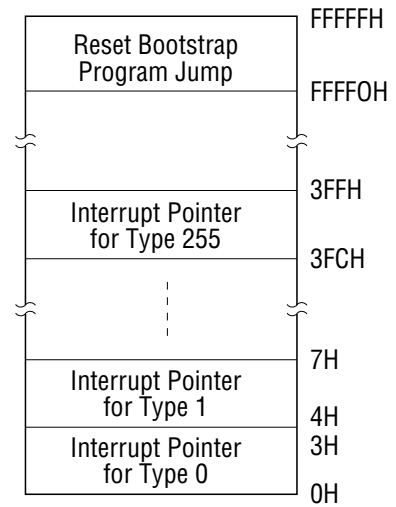
Memory location FFFF0H is the start address after reset, and 00000H through 003FFH are reserved as an interrupt pointer, where there are 256 types of interrupt pointers.

Each interrupt type has a 4-byte pointer element consisting of a 16-bit segment address and a 16-bit offset address.

Memory Organization



Reserved Memory Locations



Memory Reference Need	Segment Register Used	Segment Selection Rule
Instructions	CODE (CS)	Automatic with all instruction prefetch.
Stack	STACK (CS)	All stack pushes and pops. Memory references relative to BP base register except data references.
Local Data	DATA (DS)	Data references when relative to stack, destination of string operation, or explicitly overridden.
External (Global Data)	EXTRA (ES)	Destination of string operations: Explicitly selected using a segment overridden.

Minimum and Maximum Modes

The MSM80C86A-10 has two system modes: minimum and maximum. When using maximum mode, it is easy to organize a multi-CPU system with a MSM82C88-2 Bus Controller which generates the bus control signal.

When using minimum mode, it is easy to organize a simple system by generating bus control signal by itself.

MN/ \overline{MX} is the mode select pin. Definition of 24-31 pin changes depend on the MN/ \overline{MX} pin.

Bus Operation

The MSM80C86A-10 has a time multiplexed address and data bus. If a non-multiplexed bus is desired for a system, it is only to add the address latch.

A CPU bus cycle consists of at least four clock cycles: T1, T2, T3 and T4. (Fig. 4)

The address output occurs during T1 and data transfer occurs during T3 and T4. T2 is used for changing the direction of the bus at the read operation. When the device which is accessed by the CPU is not ready for The data transfer and the CPU "NOT READY", TW cycles are inserted between T3 and T4.

When a bus cycle is not needed, T1 cycles are inserted between the bus cycles for internal execution. During the T1 cycle, the ALE signal is output from the CPU or the MSM82C88-2 depending on MN/ \overline{MX} . At the trailing edge of ALE, a valid address may be latched.

Status bits $\overline{S_0}$, $\overline{S_1}$ and $\overline{S_2}$ are used in the maximum mode by the bus controller to recognize the type of bus operation according to the following table.

Status bits S_3 through S_7 are multiplexed with $A_{16} - A_{19}$, and \overline{BHE} : therefore, they are valid during T2 through T4.

S_3 and S_4 indicate which segment register was selected on the bus cycle, according to the following table.

$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	Characteristics
0 (LOW)	0	0	Interrupt acknowledge
0	0	1	Read I/O
0	1	0	Write I/O
0	1	1	Halt
1 (HIGH)	0	0	Instrucion Fetch
1	0	1	Read Data from Memory
1	1	0	Write Data to Memory
1	1	1	Passive (no bus cycle)

S_4	S_3	Characteristics
0 (LOW)	0	Alternate Data (Extra segment)
0	1	Stack
1 (HIGH)	0	Code or None
1	1	Data

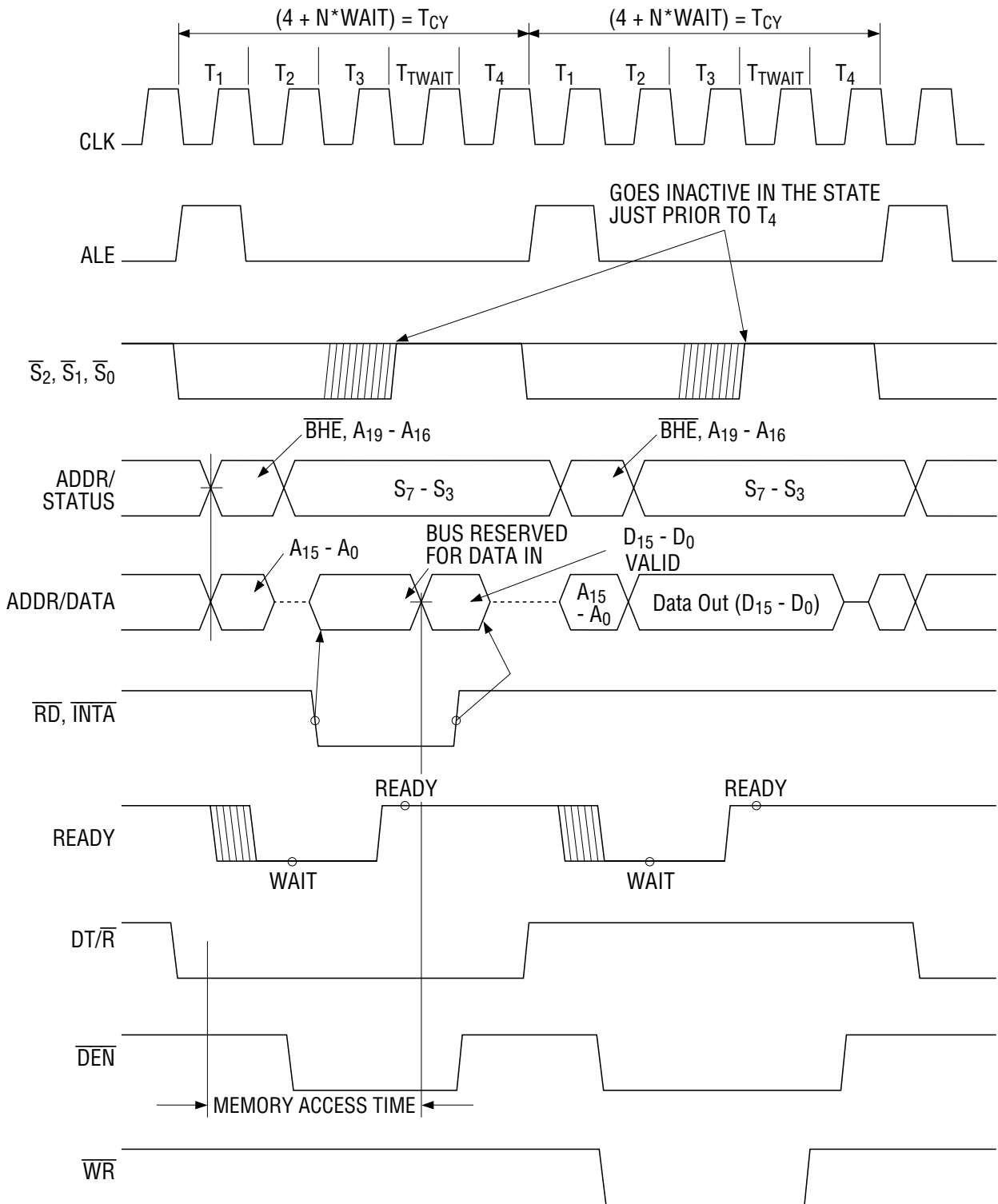
S_5 indicates interrupt enable Flag.

I/O Addressing

The MSM80C86A-10 has 64 Kbytes of I/O or as 32 Kwords I/O. When the CPU accesses an I/O device, addresses $AD_0 - AD_{15}$ are in the same format as a memory address, and $A_{16} - A_{19}$ are low.

The I/O ports addresses are same as memory, so it is necessary to be careful when using 8-bit peripherals.

Basic System Timing



EXTERNAL INTERFACE

Reset

CPU Initialization is executed by the RESET pin. The MSM80C86A-10's RESET High signal is required for greater than 4 clock cycles.

The Rising edge of RESET terminates present operation immediately. The Falling edge of RESET triggers an internal reset sequence for approximately 10 clock cycles. After the internal reset sequence is finished normal operation occurs from absolute location FFFF0H.

Interrupt Operations

Interrupt operation is classified as software or hardware, and hardware interrupt is classified as non-maskable or maskable.

An interrupt causes a new program location defined on the interrupt pointer table, according to the interrupt type. Absolute locations 00000H through 003FFH are reserved for the interrupt pointer table. The interrupt pointer table consists of 256-elements. Each element is 4 bytes in size and corresponds to an 8-bit type number which is sent from an interrupt request device during the interrupt acknowledge cycle.

Non-maskable Interrupt (NMI)

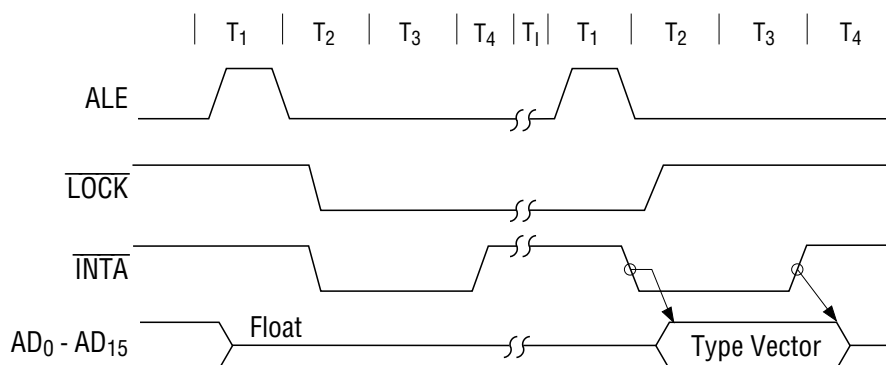
The MSM80C86A-10 has a Non-maskable interrupt (NMI) which is of higher priority than the maskable interrupt request (INTR).

The NMI request pulse width needs a minimum of 2 clock cycles. The NMI will be serviced at the end of the current instruction or between string manipulations.

Maskable Interrupt (INTR)

The MSM80C86A-10 provides another interrupt request (INTR) which can be masked by software. INTR is level triggered, so it must be held until the interrupt request is acknowledged. INTR will be serviced at the end of the current instruction or between string manipulations.

Interrupt Acknowledge Sequence



Interrupt Acknowledge

During the interrupt acknowledge sequence, further interrupts are disabled. The interrupt enable bit is reset by any interrupt, after which the Flag register is automatically pushed onto the stack. During the acknowledge sequence, the CPU emits the lock signal from T2 of the first bus cycle to T2 of the second bus cycle. At second bus cycles, byte is fetched from the external device as a vector which identified the type of interrupt. This vector is multiplied by four and used as a interrupt pointer address. (INTR only)

The interrupt Return (IRET) instruction includes a Flag pop operation which returns the original interrupt enable bit when it restores the Flag.

HALT

When a Halt instruction is executed, the CPU enters the Halt state. An interrupt request or RESET will force the MSM80C86A-10 out of the Halt state.

System Timing – Minimum Mode

A bus cycle begins T1 with an ALE signal. The trailing edge of ALE is used to latch the address. From T1 to T4 the M/\overline{IO} signal indicates a memory or I/O operation. From T2 to T4, the address data bus changes the address but to data bus.

The read (\overline{RD}), write (\overline{WR}) and interrupt acknowledge (\overline{INTA}) signals causes the addressed device to enable data bus. These signal becomes active at the beginning of T2 and inactive at the beginning of T4.

System Timing – Maximum Mode

At maximum mode, the MSM82C88-2 Bus Controller is added to system. The CPU sends status information to the Bus Controller. Bus timing signals are generated by Bus Controller. Bus timing is almost the same as in the minimum mode.

BUS HOLD CIRCUITRY

To avoid high current conditions caused by floating inputs to CMOS devices and to eliminate the need for pull-up/down resistors, "bus-hold" circuitry has been used on MSM80C86A-10 pins 2-16, 26-32, and 34-39 (Figures 6a, 6b). These circuits will maintain the last valid logic state if no driving source is present (i.e. an unconnected pin or a driving source which goes to a high impedance state). To overdrive the "bus hold" circuits, an external driver must be capable of supplying approximately 600 μ A minimum sink or source current at valid input voltage levels. Since this "bus hold" circuitry is active and not a "resistive" type element, the associated power supply current is negligible and power dissipation is significantly reduced when compared to the use of passive pull-up resistors.

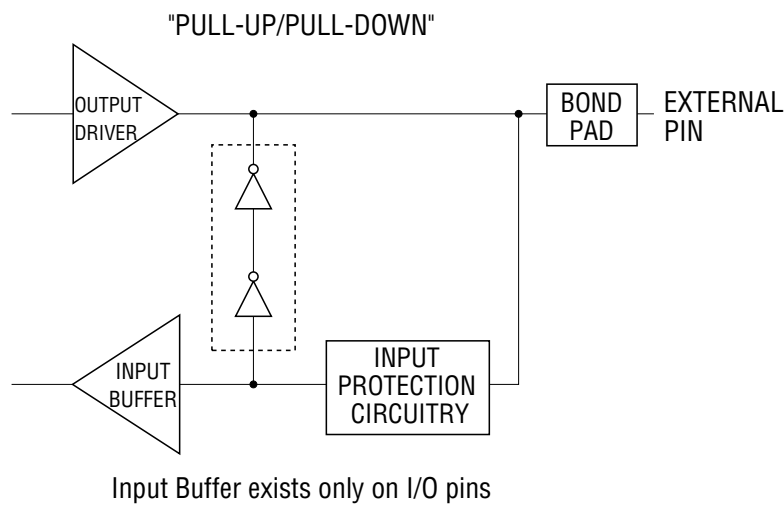


Figure 6a. Bus Hold Circuitry Pin 2-16, 35-39

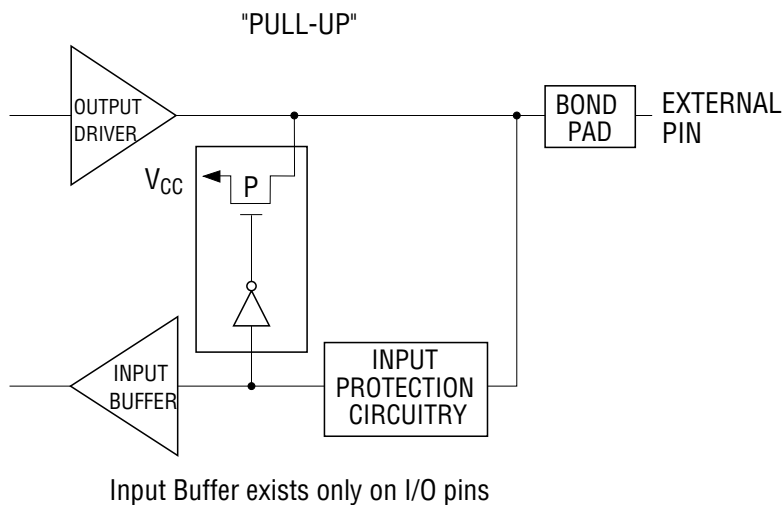


Figure 6b. Bus Hold Circuitry Pin 26-32, 34

DATA TRANSFER

MOV = Move: Register/memory to/from register Immediate to register/memory Immediate to register Memory to accumulator Accumulator to memory Register/memory to segment register Segment register to register/memory	7 1 1 1 1 1 1	6 0 1 0 0 0 0	5 0 0 1 0 0 0	4 0 0 1 0 0 0	3 1 0 0 0 1 1	2 0 1 0 0 1 0	1 d 1 reg 0 0 1	0 w w w w 0 0	7 mod mod data addr-low addr-low 0 reg 0 reg	6 0 0 0 0 0 0	5 0 0 0 0 0 0	4 reg 0 data addr-high addr-high 0 reg 0 reg	3 0 0 0 0 0 0	2 r/m r/m r/m r/m r/m r/m	1 0	0 data if w = 1 data if w = 1	
PUSH = Push: Register/memory Register Segment register	7 1 0 0	6 1 1 0	5 1 0 0	4 1 1 0	3 1 1 reg	2 1 1 1	1 1 1 0	0 1 1 0	7 mod mod mod mod	6 1 1 1	5 1 1 0	4 1 1 0	3 r/m r/m r/m				
POP = Pop: Register/memory Register Segment register	7 1 0 0	6 0 1 0	5 0 0 1	4 0 1 1	3 1 1 reg	2 1 1 1	1 1 1 0	0 reg 1 1	7 mod mod mod mod	6 0 0 0	5 0 0 0	4 0 0 0	3 r/m r/m r/m				
XCHG = Exchange: Register/memory with register Register with accumulator	7 1 1	6 0 0	5 0 0	4 0 0	3 0 1	2 1 1	1 1 reg	0 1 1	7 mod mod	6 1 1	5 0 0	4 reg reg	3 r/m r/m				
IN = Input from: Fixed port Variable port	7 1 1	6 1 1	5 1 1	4 0 0	3 0 1	2 1 0	1 0 0	0 w w	7 port port	6 1 1	5 0 0	4 port port					
OUT = Output to: Fixed port Variable port XLAT = Translate byte to AL LEA = Load EA to register LDS = Load pointer to DS LES = Load pointer to ES LAHF = Load AH with flags SAHF = Store AH into flags PUSHF = Push flags POPF = Pop flags	7 1 1 1 1 1 1 1 1 1 1	6 1 1 0 0 0 0 0 0 0 0	5 1 1 0 0 0 0 0 0 0 0	4 0 0 1 0 0 0 0 0 0 0	3 0 1 0 1 1 1 1 1 1 1	2 1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1 1	0 w w 1 1 1 1 1 1 1 0 0 0	7 mod mod mod mod mod mod mod mod mod mod mod	6 1 1 1 1 1 1 1 1 1 1	5 1 1 0 0 0 0 0 0 0 0	4 1 1 0 0 0 0 0 0 0 0	3 port port reg reg reg reg reg reg reg reg reg				

ARITHMETIC

ADD = Add: Reg./memory with register to either Immediate to register/memory Immediate to accumulator	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0	d w s w w	mod mod data	reg 0 0 0 data	r/m r/m	data data if w = 1	data if s:w = 01
ADC = Add with carry: Reg./memory with register to either Immediate to register/memory Immediate to accumulator	0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0	d w s w w	mod mod data	reg 0 1 0 data	r/m r/m	data data if w = 1	data if s:w = 01
INC = Increment: Register/memory Register AAA = ASCII adjust for add DAA = Decimal adjust for add	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1	w reg 1 1 1 1	mod 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	r/m		
SUB = Subtract: Reg./memory with register to either Immediate from register/memory Immediate from accumulator	0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1 0 0 0 0 0 0	d w s w w	mod mod data	reg 1 0 1 data	r/m r/m	data data if w = 1	data if s:w = 01
SBB = Subtract with borrow: Reg./memory with register to either Immediate from register/memory Immediate from accumulator	0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0	d w s w w	mod mod data	reg 0 1 1 data	r/m r/m	data data if w = 1	data if s:w = 01
DEC = Decrement: Register/memory Register NEG = Change sign	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 0 1 1 0 1 1 0 1 1	w reg w	mod 0 0 1 mod	0 0 1 0 1 1 0 1 1	r/m r/m		
CMP = Compare: Register/memory and register Immediate with register/memory Immediate with accumulator AAS = ASCII adjust for subtract	0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1	d w s w w w	mod mod data	reg 1 1 1 data	r/m r/m	data data if w = 1	data if s:w = 01

DAS = Decimal adjust for subtract MUL = Multiply (unsigned) IMUL = Integer multiply (signed) AAM = ASCII adjust for multiply DIV = Divide (unsigned) IDIV = Integer divide (signed) AAD = ASCII adjust for divide CBW = Convert byte to word CWD = Convert word to double word	0 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 0 1 1 1 0 0 1 1 0 0 0 0 1 0 0 1 1 1 0 0 1 1 0 0 0 1	mod 1 0 0 mod 1 0 1 0 0 0 0 1 0 1 0 mod 1 1 0 mod 1 1 1 0 0 0 0 1 0 1 0	r/m r/m 0 1 0 r/m r/m 0 1 0
--	---	--	--

LOGIC

NOT = Invert	1 1 1 1 0 0 1 1	1 1 0 1 1 0 0 0	1 1 0 0 1 1	1 1 0 1	mod	0 1 0	r/m		
SHL/SAL = Shift logical/arithmetic left	1 1 0 1 0 0 0 0	1 1 0 1 0 0 0 0	1 1 0 1 0 0	1 1 0 1	mod	1 0 0	r/m		
SHR = Shift logical right	1 1 0 1 0 0 0 0	1 1 0 1 0 0 0 0	1 1 0 1 0 0	1 1 0 1	mod	1 0 1	r/m		
SAR = Shift arithmetic right	1 1 0 1 0 0 0 0	1 1 0 1 0 0 0 0	1 1 0 1 0 0	1 1 0 1	mod	1 1 1	r/m		
ROL = Rotate left	1 1 0 1 0 0 0 0	1 1 0 1 0 0 0 0	1 1 0 1 0 0	1 1 0 1	mod	0 0 0	r/m		
ROR = Rotate right	1 1 0 1 0 0 0 0	1 1 0 1 0 0 0 0	1 1 0 1 0 0	1 1 0 1	mod	0 0 1	r/m		
RCL = Rotate left through carry	1 1 0 1 0 0 0 0	1 1 0 1 0 0 0 0	1 1 0 1 0 0	1 1 0 1	mod	0 1 0	r/m		
RCR = Rotate right through carry	1 1 0 1 0 0 0 0	1 1 0 1 0 0 0 0	1 1 0 1 0 0	1 1 0 1	mod	0 1 1	r/m		
AND = And: Reg./memory and register to either Immediate to register/memory Immediate to accumulator	0 0 1 0 0 0 0 0	0 0 1 0 0 0 0 0	0 0 1 0 0 0	0 0 1 0	mod	reg	r/m		
	1 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0	1 0 0 0 0 0	1 0 0 0	mod	1 0 0	r/m		data if w = 1
	0 0 1 0 0 0 1 0	0 0 1 0 0 0 1 0	0 0 1 0 0 1	0 0 1 0	mod	data	r/m		
TEST = And function to flags, no result: Register/memory and register Immediate data and register/memory Immediate data and accumulator	1 0 0 0 0 1 0 0	1 1 1 1 0 1 1 0	1 0 1 0 1 0 0 0	1 0 1 0	mod	reg	r/m		
	1 1 1 1 0 1 1 0	1 1 1 1 0 1 1 0	1 1 1 1 0 1	1 1 1 1	mod	0 0 0	r/m		data if w = 1
	1 0 1 0 1 0 0 0	1 0 1 0 1 0 0 0	1 0 1 0 1 0	1 0 1 0	mod	data	r/m		
OR = Or: Reg./memory and register to either Immediate to register/memory Immediate to accumulator	0 0 0 0 0 1 0 0	1 0 0 0 0 0 0 0	0 0 0 0 1 1 0 0	0 0 0 0	mod	reg	r/m		
	1 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0	1 0 0 0 0 0	1 0 0 0	mod	0 0 1	r/m		data if w = 1
	0 0 0 0 1 1 0 0	0 0 0 0 1 1 0 0	0 0 0 0 1 1	0 0 0 0	mod	data	r/m		
XOR = Exclusive or: Reg./memory and register to either Immediate to register/memory Immediate to accumulator	0 0 1 1 0 0 0 0	1 0 0 0 0 0 0 0	0 0 1 1 0 0 1 0	0 0 1 1	mod	reg	r/m		
	1 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0	1 0 0 0 0 0	1 0 0 0	mod	1 1 0	r/m		data if w = 1
	0 0 1 1 0 0 1 0	0 0 1 1 0 0 1 0	0 0 1 1 0 1	0 0 1 1	mod	data	r/m		

STRING MANIPULATION

REP = Repeat	1	1	1	1	1	1	1	1	0	0	1	z	
MOVS = Move byte/word	1	0	1	0	0	1	0	w					
CMPS = Compare byte/word	1	0	1	0	0	1	1	w					
SCAS = Scan byte/word	1	0	1	0	1	1	1	w					
LODS = Load byte/word to AL/AX	1	0	1	0	1	1	0	w					
STOS = Store byte/word from AL/AX	1	0	1	0	1	0	1	w					
CJMP = Conditional JMP													
JE/JZ = Jump on equal/zero	0	1	1	1	0	1	0	0	disp				
JZ/JNGE = Jump on less/not greater or equal	0	1	1	1	1	1	0	0	disp				
JLE/JJNG = Jump on less or equal/not greater	0	1	1	1	1	1	1	0	disp				
JB/JNAE = Jump on below/not above or equal	0	1	1	1	0	0	1	0	disp				
JBE/JNA = Jump on below or equal/not above	0	1	1	1	0	1	1	0	disp				
JP/JPE = Jump on parity/parity even	0	1	1	1	1	0	1	0	disp				
JO = Jump on over flow	0	1	1	1	0	0	0	0	disp				
JS = Jump on sign	0	1	1	1	1	0	0	0	disp				
JNE/JNZ = Jump on not equal/not zero	0	1	1	1	0	1	0	1	disp				
JNL/JGE = Jump on not less/greater or equal	0	1	1	1	1	1	0	1	disp				
JNLE/JG = Jump on not less or equal/greater	0	1	1	1	1	1	1	1	disp				
JNB/JAE = Jump on not below/above or equal	0	1	1	1	0	0	1	1	disp				
JNBE/JA = Jump on not below or equal/above	0	1	1	1	0	1	1	1	disp				
JNP/JPO = Jump on not parity/parity odd	0	1	1	1	1	0	1	1	disp				
JNO = Jump on not overflow	0	1	1	1	1	0	0	1	disp				
JNS = Jump on not sign	0	1	1	1	1	0	0	1	disp				
LOOP = Loop CX times	1	1	1	1	0	0	1	0	disp				
LOOPZ/LOOPE = Loop while zero/equal	1	1	1	0	0	0	0	1	disp				
LOOPNZ/LOOPNE = Loop while not zero equal	1	1	1	0	0	0	0	0	disp				
JCXZ = Jump on CX zero	1	1	1	0	0	0	1	1	disp				
INT = Interrupt													
Type specified	1	1	0	0	1	1	0	1	type				
Type 3	1	1	0	0	1	1	0	0					
INTO = Interrupt on overflow	1	1	0	0	1	1	1	0					
IRET = Interrupt return	1	1	0	0	1	1	1	1					

PROCESSOR CONTROL

CLC = Clear carry	1 1 1 1 1 0 0 0			
CMC = Complementary carry	1 1 1 1 0 1 0 1			
STC = Set carry	1 1 1 1 1 0 0 1			
CLD = Clear direction	1 1 1 1 1 1 0 0			
STD = Set direction	1 1 1 1 1 1 0 1			
CLI = Clear interrupt	1 1 1 1 1 0 1 0			
STI = Set interrupt	1 1 1 1 1 0 1 1			
HLT = Halt	1 1 1 1 0 1 0 0			
WAIT = Wait	1 0 0 1 1 0 1 1			
ESC = Escape (to external device)	1 1 0 1 1 x x x	mod	x x x	r/m
LOCK = Bus lock prefix	1 1 1 1 0 0 0 0			

CONTROL TRANSFER

CALL = Call:	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Direct within segment	1 1 1 0 1 0 0 0	disp-low		disp-high
Indirect within segment	1 1 1 1 1 1 1 1	mod 0 1 0	r/m	
Direct intersegment	1 0 0 1 1 0 1 0	offset-low		offset-high
Indirect intersegment	1 1 1 1 1 1 1 1	mod 0 1 1	r/m	seg-high
JMP = Unconditional Jump:				
Direct within segment	1 1 1 0 1 0 0 1	disp-low		disp-high
Direct within segment-short	1 1 1 0 1 0 1 1	disp		
Indirect within segment	1 1 1 1 1 1 1 1	mod 1 0 0	r/m	
Direct intersegment	1 1 1 0 1 0 1 0	offset-low		offset-high
Indirect intersegment	1 1 1 1 1 1 1 1	mod 1 0 1	r/m	seg-high
RET = Return from CALL:				
Within segment	1 1 0 0 0 0 1 1			
Within seg. adding immediate to SP	1 1 0 0 0 0 1 0	data-low		data-high
Intersegment	1 1 0 0 1 0 1 1			
Intersegment adding immediate to SP	1 1 0 0 1 0 1 0	data-low		dat-high

Foot Notes: AL = 8-bit accumulator

AX = 18-bit accumulator

CX = Count register

DS = Data segment

ES = Extra segment

Above/below refers to unsigned value

Greater=more positive

Less=less positive (more negative) signed value

If d=1 then "to" reg: If d=0 then "from" reg.

If w=1 then word instruction: If w=0 then byte instruction

If mod=11 then r/m is treated as a REG field

If mod=00 then DISP=0*, disp-low and disp-high are absent

If mod=01 then DISP=disp-low sign-extended to 16 bits, disp-high is absent

If mod=10 then DISP=disp-high: disp-low

If r/m=000 then EA=(BX)+(SI)+DISP

If r/m=001 then EA=(BX)+(DI)+DISP

If r/m=010 then EA=(BP)+(SI)+DISP

If r/m=011 then EA=(BP)+(DI)+DISP

If r/m=100 then EA=(SI)+DISP

If r/m=101 then EA=(DI)+DISP

If r/m=110 then EA=(BP)+DISP*

If r/m=111 then EA=(BX)+DISP

DISP follows 2nd byte of instruction (before data if required)

* except if mod=00 and r/m=110 then EA-disp-high: disp-low

If s:w=01 then 16 bits of immediate data form the operand

If s:w=11 then an immediate data byte is sign extended to form the 16-bit operand

If v=0 then "count"=1:if v=1 then "count" in (CL)

x=don' t care

z is used for string primitives for comparison with ZF FLAG

SEGMENT OVERRIDE PREFIX

001 reg 110

REG is assigned according to the following table:

16-Bit (w=1)	8-Bit (w=0)	Segment
000 AX	000 AL	00 ES
001 CX	001 CL	01 CS
010 DX	010 DL	10 SS
011 BX	011 BL	11 DS
100 SP	100 AH	
101 BP	101 CH	
110 SI	110 DH	
111 DI	111 BH	

Instructions which reference the flag register file as a 16-bit object use the symbol FLAGS to represent the file:

FLAGS=x:x:x:x:(OF):(DF):(IF):(TF):(SF):(ZF):X:(AF):X:(PF):X:(CF)

NOTICE ON REPLACING LOW-SPEED DEVICES WITH HIGH-SPEED DEVICES

The conventional low speed devices are replaced by high-speed devices as shown below. When you want to replace your low speed devices with high-speed devices, read the replacement notice given on the next pages.

High-speed device (New)	Low-speed device (Old)	Remarks
M80C85AH	M80C85A/M80C85A-2	8bit MPU
M80C86A-10	M80C86A/M80C86A-2	16bit MPU
M80C88A-10	M80C88A/M80C88A-2	8bit MPU
M82C84A-2	M82C84A/M82C84A-5	Clock generator
M81C55-5	M81C55	RAM.I/O, timer
M82C37B-5	M82C37A/M82C37A-5	DMA controller
M82C51A-2	M82C51A	USART
M82C53-2	M82C53-5	Timer
M82C55A-2	M82C55A-5	PPI

Differences between MSM80C86A-10 and MSM80C86A-2, MSM80C86A**1) Manufacturing Process**

All devices use a 1.5 μ Si-CMOS process technology.

2) Design

Although circuit timings of these devices are a little different, these devices have the same chip size and logics.

3) Electrical Characteristics

Oki's '96 Data Book for MICROCONTROLLER describes that the MSM80C86A-10 satisfies the electrical characteristics of the MSM80C86A-2 and MSM80C86A.

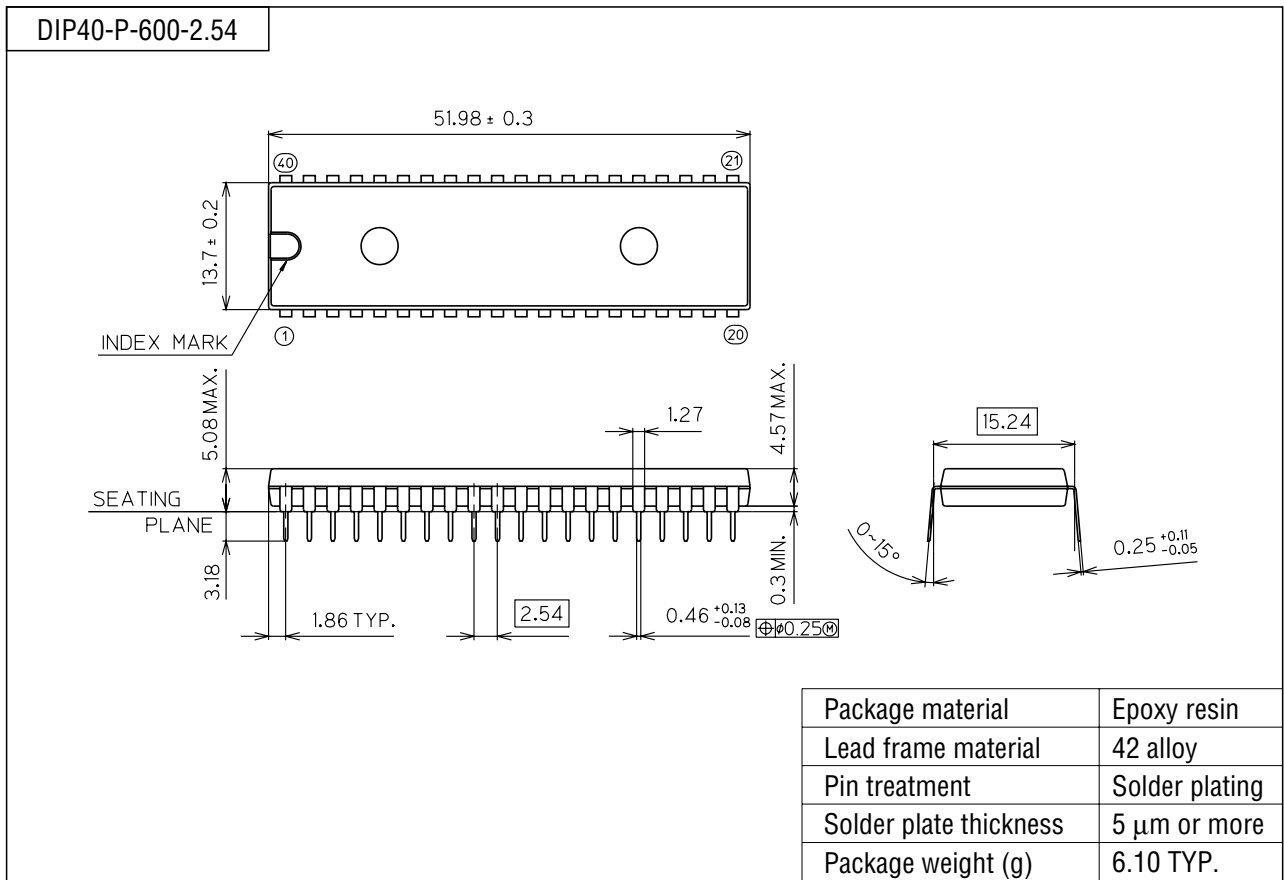
4) Other notices

1) The noise characteristics of the high-speed MSM80C86A-10 (for 10 MHz) are a little different from those of the MSM80C86A-2 and MSM80C86A. Therefore when devices are replaced for upgrading, it is recommended to perform noise evaluation.

2) The characteristics of the MSM80C86A-10 basically satisfy those of the MSM80C86A-2 and MSM80C86A but their timings are a little different. When critical timing is required in designing it is recommended to evaluate operating margins at various temperatures and voltages.

PACKAGE DIMENSIONS

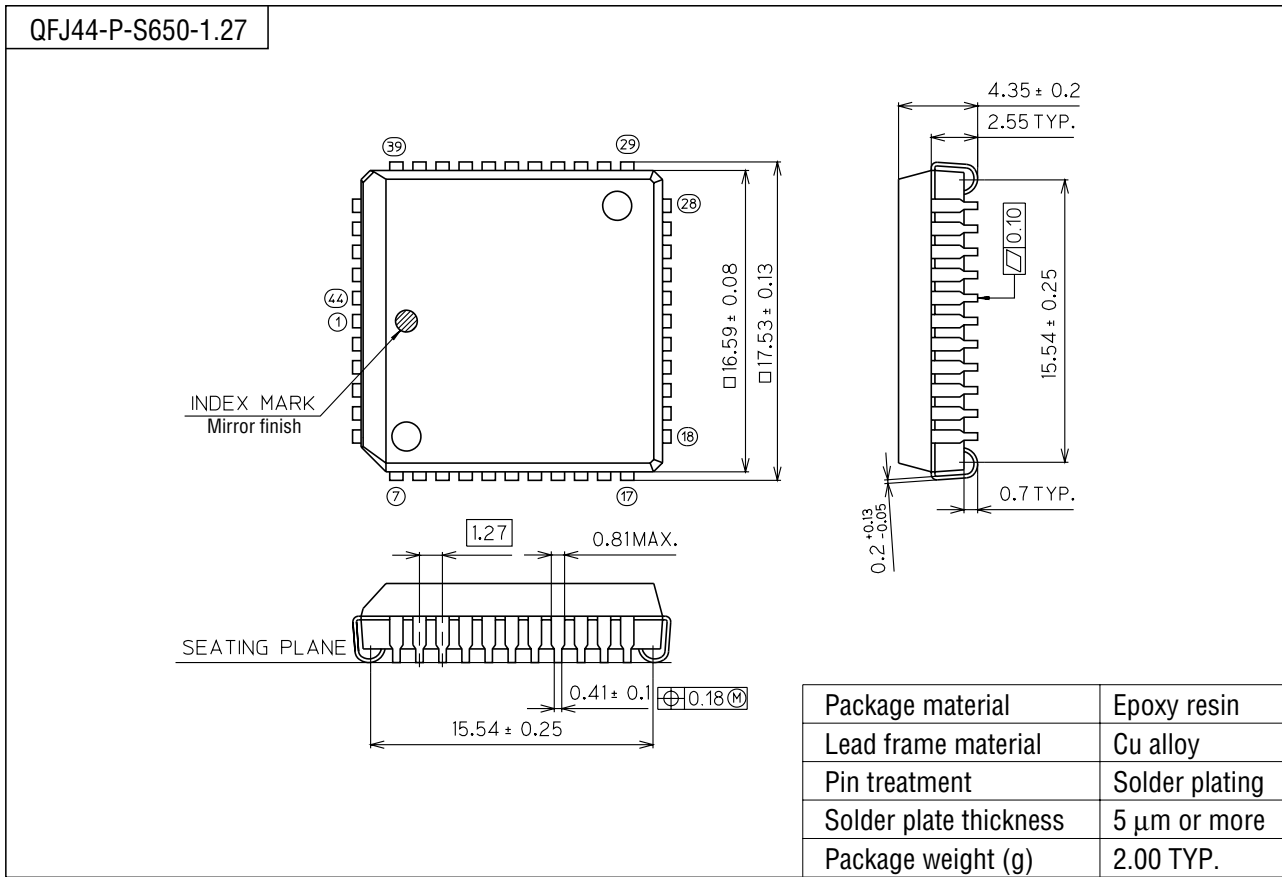
(Unit : mm)



Notes for Mounting the Surface Mount Type Package

The SOP, QFP, TSOP, SOJ, QFJ (PLCC), SHP and BGA are surface mount type packages, which are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact Oki's responsible sales person for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

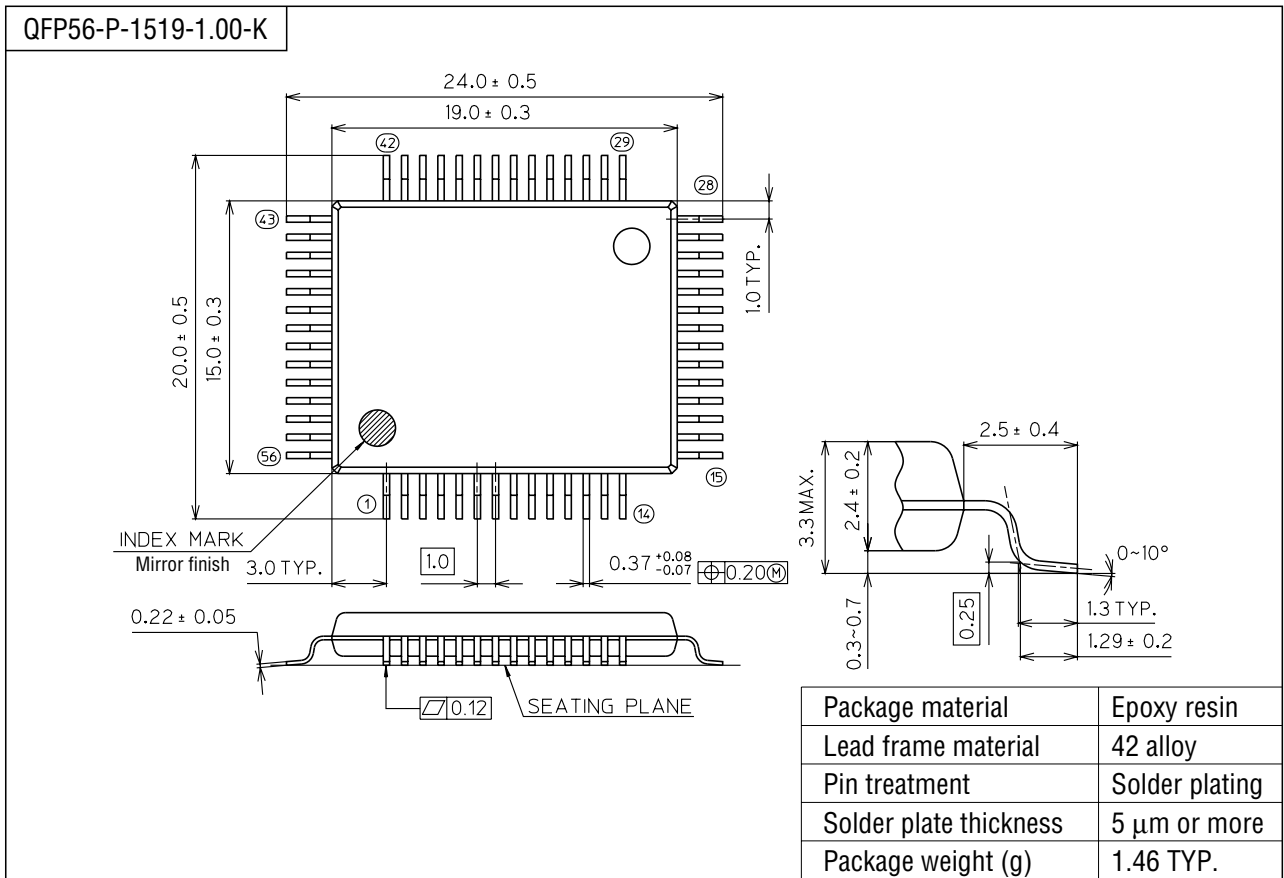
(Unit : mm)



Notes for Mounting the Surface Mount Type Package

The SOP, QFP, TSOP, SOJ, QFJ (PLCC), SHP and BGA are surface mount type packages, which are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact Oki's responsible sales person for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

(Unit : mm)



Notes for Mounting the Surface Mount Type Package

The SOP, QFP, TSOP, SOJ, QFJ (PLCC), SHP and BGA are surface mount type packages, which are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact Oki's responsible sales person for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).