

## Radiation Hardened 16-Bit CMOS Microprocessor

The Intersil HS-80C86RH high performance radiation hardened 16-bit CMOS CPU is manufactured using a hardened field, self aligned silicon gate CMOS process. Two modes of operation, MINimum for small systems and MAXimum for larger applications such as multiprocessing, allow user configuration to achieve the highest performance level. Industry standard operation allows use of existing NMOS 8086 hardware and software designs.

**Specifications for Rad Hard QML devices are controlled by the Defense Supply Center in Columbus (DSCC). The SMD numbers listed here must be used when ordering.**

**Detailed Electrical Specifications for these devices are contained in SMD 5962-95722. A “hot-link” is provided on our homepage for downloading.**  
[www.intersil.com/spacedefense/space.asp](http://www.intersil.com/spacedefense/space.asp)

### Ordering Information

ORDERING NUMBER	INTERNAL MKT. NUMBER	TEMP. RANGE (°C)
5962R9572201QQC	HS1-80C86RH-8	-55 to 125
5962R9572201QXC	HS9-80C86RH-8	-55 to 125
5962R9572201VQC	HS1-80C86RH-Q	-55 to 125
5962R9572201VXC	HS9-80C86RH-Q	-55 to 125
HS1-80C86RH/Proto	HS1-80C86RH/Proto	-55 to 125
HS9-80C86RH/Proto	HS9-80C86RH/Proto	-55 to 125

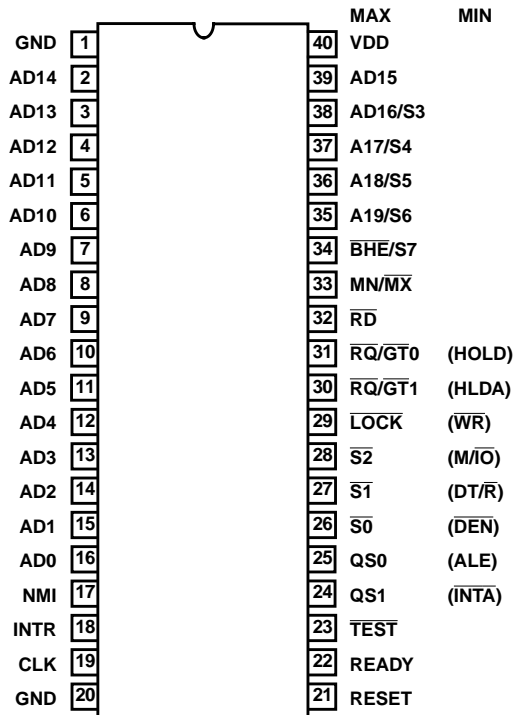
### Features

- Electrically Screened to SMD # 5962-95722
- QML Qualified per MIL-PRF-38535 Requirements
- Radiation Performance
  - Latch Up Free EPI-CMOS
  - Total Dose . . . . . 100 krad(Si) (Max)
  - Transient Upset . . . . . >10<sup>8</sup> rad(Si)/s
- Low Power Operation
  - ICCSB . . . . . 500µA (Max)
  - ICCOP . . . . . 12mA/MHz (Max)
- Pin Compatible with NMOS 8086 and Intersil 80C86
- Completely Static Design DC to 5MHz
- 1MB Direct Memory Addressing Capability
- 24 Operand Addressing Modes
- Bit, Byte, Word, and Block Move Operations
- 8-Bit and 16-Bit Signed/Unsigned Arithmetic
  - Binary or Decimal
  - Multiply and Divide
- Bus-Hold Circuitry Eliminates Pull-up Resistors for CMOS Designs
- Hardened Field, Self-Aligned, Junction-Isolated CMOS Process
- Single 5V Power Supply
- Military Temperature Range . . . . . -35°C to 125°C
- Minimum LET for Single Event Upset . . . . . 6MEV/mg/cm<sup>2</sup> (Typ)

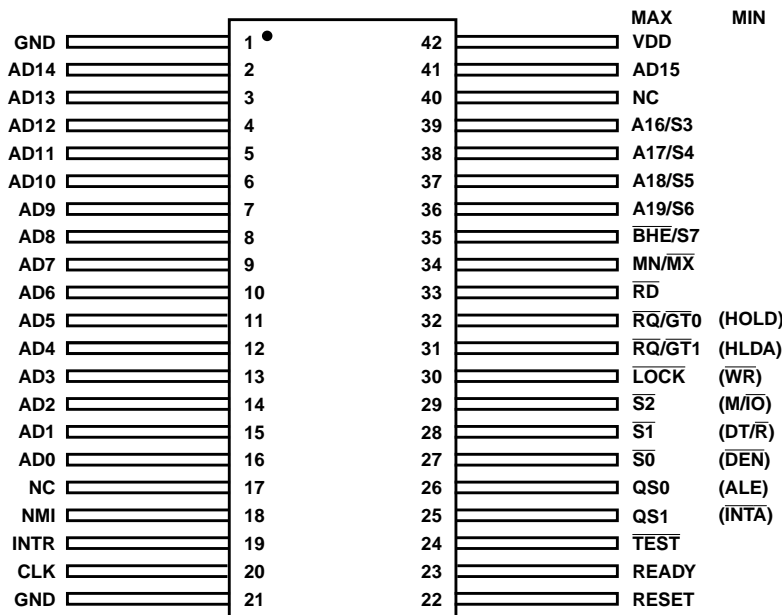
# HS-80C86RH

## Pinout

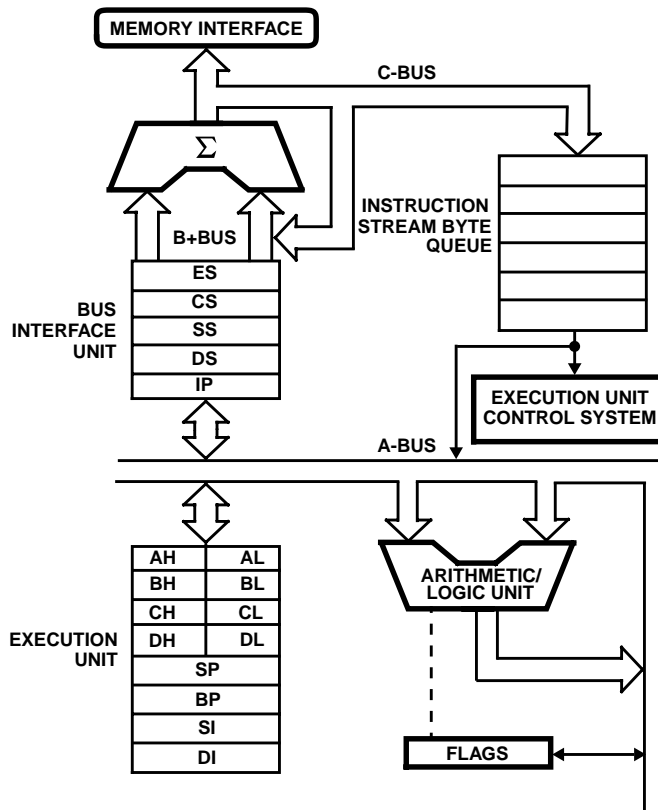
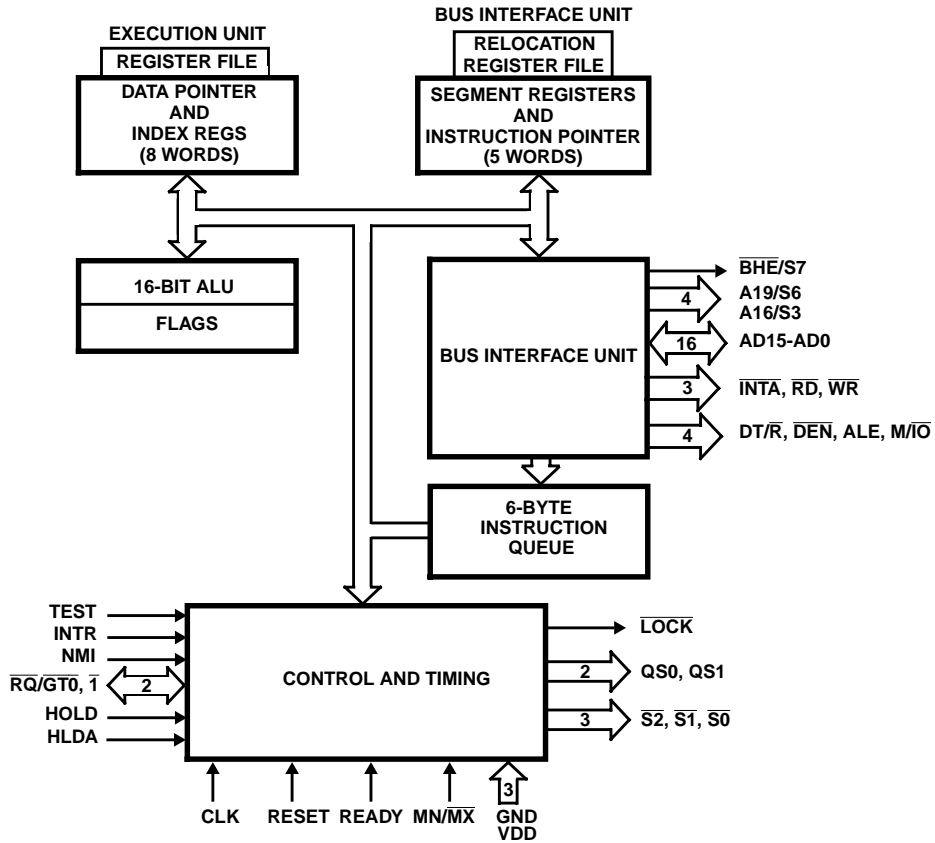
HS-80C86RH 40 LEAD CERAMIC DUAL-IN-LINE METAL SEAL PACKAGE (SBDIP)  
MIL-STD-1835, CDIP2-T40  
TOP VIEW



HS-80C86RH 42 LEAD CERAMIC METAL SEAL FLATPACK PACKAGE (FLATPACK)  
INTERSIL OUTLINE K42.A  
TOP VIEW



Functional Diagram



## HS-80C86RH

### Pin Descriptions

SYMBOL	PIN NUMBER	TYPE	DESCRIPTION															
<p>The following pin function descriptions are for HS-80C86RH systems in either minimum or maximum mode. The "Local Bus" in these descriptions is the direct multiplexed bus interface connection to the HS-80C86RH (without regard to additional bus buffers).</p>																		
AD15-AD0	2-16, 39	I/O	<p>ADDRESS DATA BUS: These lines constitute the time multiplexed memory/I/O address (T1) and data (T2, T3, TW, T4) bus. AO is analogous to <math>\overline{\text{BHE}}</math> for the lower byte of the data bus, pins D7-D0. It is LOW during T1 when a byte is to be transferred on the lower portion of the bus in memory or I/O operations. Eight-bit oriented devices tied to the lower half would normally use AD0 to condition chip select functions (See <math>\overline{\text{BHE}}</math>). These lines are active HIGH and are held at high impedance to the last valid logic level during interrupt acknowledge and local bus "hold acknowledge" or "grant sequence".</p>															
A19/S6 A18/S5 A17/S4 A16/S3	35-38	O	<p>ADDRESS/STATUS: During T1, these are the four most significant address lines for memory operations. During I/O operations these lines are low. During memory and I/O operations, status information is available on these lines during T2, T3, TW, T4. S6 is always zero. The status of the interrupt enable FLAG bit (S5) is updated at the beginning of each CLK cycle. S4 and S3 are encoded.</p> <p>This information indicates which segment register is presently being used for data accessing. These lines are held at high impedance to the last valid logic level during local bus "hold acknowledge" or "grant sequence".</p> <table style="margin-left: 40px;"> <thead> <tr> <th style="text-align: center;">S4</th> <th style="text-align: center;">S3</th> <th></th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>Extra Data</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>Stack</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>Code or None</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>Data</td> </tr> </tbody> </table>	S4	S3		0	0	Extra Data	0	1	Stack	1	0	Code or None	1	1	Data
S4	S3																	
0	0	Extra Data																
0	1	Stack																
1	0	Code or None																
1	1	Data																
$\overline{\text{BHE}}/\text{S7}$	34	O	<p>BUS HIGH ENABLE/STATUS: During T1 the bus high enable signal (<math>\overline{\text{BHE}}</math>) should be used to enable data onto the most significant half of the data bus, pins D15-D8. Eight bit oriented devices tied to the upper half of the bus would normally use <math>\overline{\text{BHE}}</math> to condition chip select functions. <math>\overline{\text{BHE}}</math> is LOW during T1 for read, write, and interrupt acknowledge cycles when a byte is to be transferred on the high portion of the bus. The S7 status information is available during T2, T3 and T4. The signal is active LOW, and is held at high impedance to the last valid logic level during interrupt acknowledge and local bus "hold acknowledge" or "grant sequence"; it is LOW during T1 for the first interrupt acknowledge cycle.</p> <table style="margin-left: 40px;"> <thead> <tr> <th style="text-align: center;"><math>\overline{\text{BHE}}</math></th> <th style="text-align: center;">A0</th> <th></th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>Whole Word</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>Upper Byte from/to Odd Address</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>Lower Byte from/to Even Address</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>None</td> </tr> </tbody> </table>	$\overline{\text{BHE}}$	A0		0	0	Whole Word	0	1	Upper Byte from/to Odd Address	1	0	Lower Byte from/to Even Address	1	1	None
$\overline{\text{BHE}}$	A0																	
0	0	Whole Word																
0	1	Upper Byte from/to Odd Address																
1	0	Lower Byte from/to Even Address																
1	1	None																
$\overline{\text{RD}}$	32	O	<p>READ: Read strobe indicates that the processor is performing a memory or I/O read cycle, depending on the state of the <math>\overline{\text{M}/\overline{\text{IO}}}</math> or <math>\overline{\text{S2}}</math> pin. This signal is used to read devices which reside on the HS-80C86RH local bus. <math>\overline{\text{RD}}</math> is active LOW during T2, T3 and TW of any read cycle, and is guaranteed to remain HIGH in T2 until the 80C86 local bus has floated.</p> <p>This line is held at a high impedance logic one state during "hold acknowledge" or "grant sequence".</p>															
READY	22	I	<p>READY: is the acknowledgment from the addressed memory or I/O device that will complete the data transfer. The RDY signal from memory or I/O is synchronized by the HS-82C85RH Clock Generator to form READY. This signal is active HIGH. The HS-80C86RH READY input is not synchronized. Correct operation is not guaranteed if the Setup and Hold Times are not met.</p>															
INTR	18	I	<p>INTERRUPT REQUEST: is a level triggered input which is sampled during the last clock cycle of each instruction to determine if the processor should enter into an interrupt acknowledge operation. If so, an interrupt service routine is called via an interrupt vector lookup table located in system memory. INTR is internally synchronized and can be internally masked by software resetting the interrupt enable bit. This signal is active HIGH.</p>															
$\overline{\text{TEST}}$	23	I	<p>TEST: input is examined by the "Wait" instruction. If the <math>\overline{\text{TEST}}</math> input is LOW execution continues, otherwise the processor waits in an "Idle" state. This input is synchronized internally during each clock cycle on the leading edge of CLK.</p>															

**Pin Descriptions** (Continued)

SYMBOL	PIN NUMBER	TYPE	DESCRIPTION
NMI	17	I	NON-MASKABLE INTERRUPT: is an edge triggered input which causes a type 2 interrupt. An interrupt service routine is called via an interrupt vector lookup table located in system memory. NMI is not maskable internally by software. A transition from LOW to HIGH initiates the interrupt at the end of the current instruction. This input is internally synchronized.
RESET	21	I	RESET: causes the processor to immediately terminate its present activity. The signal must change from LOW to HIGH and remain active HIGH for at least 4 CLK cycles. It restarts execution, as described in the Instruction Set description, when RESET returns LOW. RESET is internally synchronized.
CLK	19	I	CLOCK: provides the basic timing for the processor and bus controller. It is asymmetric with a 33% duty cycle to provide optimized internal timing.
VDD	40		VDD: +5V power supply pin. A 0.1μF capacitor between pins 20 and 40 is recommended for decoupling.
GND	1, 20		GND: Ground. Note: both must be connected. A 0.1μF capacitor between pins 1 and 20 is recommended for decoupling.
MN/MX	33	I	MINIMUM/MAXIMUM: Indicates what mode the processor is to operate in. The two modes are discussed in the following sections.

The following pin function descriptions are for the HS-80C86RH system in maximum mode (i.e., MN/MX = GND). Only the pin functions which are unique to maximum mode are described below.

$\overline{S0}, \overline{S1}, \overline{S2}$	26-28	O	<p>STATUS: is active during T4, T1 and T2 and is returned to the passive state (1,1,1) during T3 or during TW when READY is HIGH. This status is used by the 82C88 Bus Controller to generate all memory and I/O access control signals. Any change by <math>\overline{S2}</math>, <math>\overline{S1}</math>, or <math>\overline{S0}</math> during T4 is used to indicate the beginning of a bus cycle, and the return to the passive state in T3 or TW is used to indicate the end of a bus cycle. These status lines are encoded. These signals are held at a high impedance logic one state during "grant sequence".</p> <table border="1"> <thead> <tr> <th><math>\overline{S2}</math></th> <th><math>\overline{S1}</math></th> <th><math>\overline{S0}</math></th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Interrupt Acknowledge</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Read I/O Port</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Write I/O Port</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Halt</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Code Access</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Read Memory</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Write Memory</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Passive</td> </tr> </tbody> </table>	$\overline{S2}$	$\overline{S1}$	$\overline{S0}$		0	0	0	Interrupt Acknowledge	0	0	1	Read I/O Port	0	1	0	Write I/O Port	0	1	1	Halt	1	0	0	Code Access	1	0	1	Read Memory	1	1	0	Write Memory	1	1	1	Passive
$\overline{S2}$	$\overline{S1}$	$\overline{S0}$																																					
0	0	0	Interrupt Acknowledge																																				
0	0	1	Read I/O Port																																				
0	1	0	Write I/O Port																																				
0	1	1	Halt																																				
1	0	0	Code Access																																				
1	0	1	Read Memory																																				
1	1	0	Write Memory																																				
1	1	1	Passive																																				

Pin Descriptions (Continued)

SYMBOL	PIN NUMBER	TYPE	DESCRIPTION															
$\overline{RQ/GT0}$ $\overline{RQ/GT1}$	31, 30	I/O	<p>REQUEST/GRANT: pins are used by other local bus masters to force the processor to release the local bus at the end of the processor's current bus cycle. Each pin is bidirectional with <math>\overline{RQ/GT0}</math> having higher priority than <math>\overline{RQ/GT1}</math>. <math>\overline{RQ/GT}</math> has an internal pull-up bus hold device so it may be left unconnected. The request/grant sequence is as follows (see <math>\overline{RQ/GT}</math> Sequence Timing.)</p> <ol style="list-style-type: none"> <li>1. A pulse of 1 CLK wide from another local bus master indicates a local bus request ("hold") to the HS-80C86RH (pulse 1).</li> <li>2. During a T4 or T1 clock cycle, a pulse 1 CLK wide from the HS-80C86RH to the requesting master (pulse 2) indicates that the HS-80C86RH has allowed the local bus to float and that it will enter the "grant sequence" state at the next CLK. The CPU's bus interface unit is disconnected logically from the local bus during "grant sequence".</li> <li>3. A pulse 1 CLK wide from the requesting master indicates to the HS-80C86RH (pulse 3) that the "hold" request is about to end and that the HS-80C86RH can reclaim the local bus at the next CLK. The CPU then enters T4 (or T1 if no bus cycles pending).</li> </ol> <p>Each Master-Master exchange of the local bus is a sequence of 3 pulses. There must be one idle CLK cycle after each bus exchange. Pulses are active low.</p> <p>If the request is made while the CPU is performing a memory cycle, it will release the local bus during T4 of the cycle when all the following conditions are met:</p> <ol style="list-style-type: none"> <li>1. Request occurs on or before T2.</li> <li>2. Current cycle is not the low byte of a word (on an odd address).</li> <li>3. Current cycle is not the first acknowledge of an interrupt acknowledge sequence.</li> <li>4. A locked instruction is not currently executing.</li> </ol> <p>If the local bus is idle when the request is made the two possible events will follow:</p> <ol style="list-style-type: none"> <li>1. Local bus will be released during the next cycle.</li> <li>2. A memory cycle will start within 3 CLKs. Now the four rules for a currently active memory cycle apply with condition number 1 already satisfied.</li> </ol>															
$\overline{LOCK}$	29	O	<p>LOCK: output indicates that other system bus masters are not to gain control of the system bus while <math>\overline{LOCK}</math> is active LOW. The <math>\overline{LOCK}</math> signal is activated by the "LOCK" prefix instruction and remains active until the completion of the next instruction. This signal is active LOW, and is held at a HIGH impedance logic one state during "grant sequence". In MAX mode, <math>\overline{LOCK}</math> is automatically generated during T2 of the first <math>\overline{INTA}</math> cycle and removed during T2 of the second <math>\overline{INTA}</math> cycle.</p>															
QS1, QS0	24, 25	O	<p>QUEUE STATUS: The queue status is valid during the CLK cycle after which the queue operation is performed.</p> <p>QS1 and QS2 provide status to allow external tracking of the internal HS-80C86RH instruction queue. Note that QS1, QS0 never become high impedance.</p> <table border="0"> <thead> <tr> <th>QS1</th> <th>QS0</th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>No Operation</td> </tr> <tr> <td>0</td> <td>1</td> <td>First Byte of Opcode from Queue</td> </tr> <tr> <td>1</td> <td>0</td> <td>Empty the Queue</td> </tr> <tr> <td>1</td> <td>1</td> <td>Subsequent Byte from Queue</td> </tr> </tbody> </table>	QS1	QS0		0	0	No Operation	0	1	First Byte of Opcode from Queue	1	0	Empty the Queue	1	1	Subsequent Byte from Queue
QS1	QS0																	
0	0	No Operation																
0	1	First Byte of Opcode from Queue																
1	0	Empty the Queue																
1	1	Subsequent Byte from Queue																

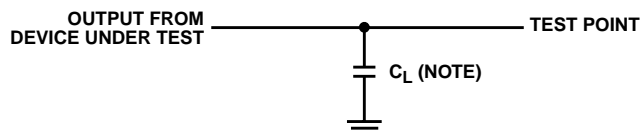
The following pin function descriptions are for the HS-80C86RH in minimum mode (i.e.,  $MN/\overline{MX} = VDD$ ). Only the pin functions which are unique to minimum mode are described; all other pin functions are as described below.

$\overline{M/\overline{IO}}$	28	O	<p>STATUS LINE: logically equivalent to <math>\overline{S2}</math> in the maximum mode. It is used to distinguish a memory access from an I/O access. <math>\overline{M/\overline{IO}}</math> becomes valid in the T4 preceding a bus cycle and remains valid until the final T4 of the cycle (M = HIGH, IO = LOW). <math>\overline{M/\overline{IO}}</math> is held to a high impedance logic zero during local bus "hold acknowledge".</p>
$\overline{WR}$	29	O	<p>WRITE: indicates that the processor is performing a write memory or write I/O cycle, depending on the state of the <math>\overline{M/\overline{IO}}</math> signal. <math>\overline{WR}</math> is active for T2, T3 and TW of any write cycle. It is active LOW, and is held to high impedance logic one during local bus "hold acknowledge".</p>
$\overline{INTA}$	24	O	<p>INTERRUPT ACKNOWLEDGE: is used as a read strobe for interrupt acknowledge cycles. It is active LOW during T2, T3 and TW of each interrupt acknowledge cycle. Note that <math>\overline{INTA}</math> is never floated.</p>

Pin Descriptions (Continued)

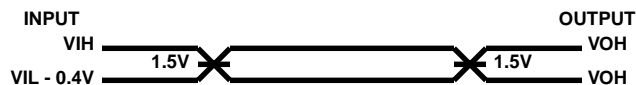
SYMBOL	PIN NUMBER	TYPE	DESCRIPTION
ALE	25	O	ADDRESS LATCH ENABLE: is provided by the processor to latch the address into the 82C82 latch. It is a HIGH pulse active during clock LOW of T1 of any bus cycle. Note that ALE is never floated.
DT/ $\bar{R}$	27	O	DATA TRANSMIT/RECEIVE: is needed in a minimum system that desires to use a data bus transceiver. It is used to control the direction of data flow through the transceiver. Logically, DT/ $\bar{R}$ is equivalent to $\bar{S1}$ in maximum mode, and its timing is the same as for M/ $\bar{I}$ O (T = HIGH, R = LOW). DT/ $\bar{R}$ is held to a high impedance logic one during local bus "hold acknowledge".
$\bar{DEN}$	26	O	DATA ENABLE: provided as an output enable for a bus transceiver in a minimum system which uses the transceiver. $\bar{DEN}$ is active LOW during each memory and I/O access and for $\bar{INTA}$ cycles. For a read or $\bar{INTA}$ cycle it is active from the middle of T2 until the middle of T4, while for a write cycle it is active from the beginning of T2 until the middle of T4. $\bar{DEN}$ is held to a high impedance logic one during local bus "hold acknowledge".
HOLD HLDA	31 30	I O	HOLD: indicates that another master is requesting a local bus "hold". To be acknowledged, HOLD must be active HIGH. The processor receiving the "hold" will issue a "hold acknowledge" (HLDA) in the middle of a T4 or T1 clock cycle. Simultaneously with the issuance of HLDA, the processor will float the local bus and control lines. After HOLD is detected as being LOW, the processor will lower HLDA, and when the processor needs to run another cycle, it will again drive the local bus and control lines.  HOLD is not an asynchronous input. External synchronization should be provided if the system cannot otherwise guarantee the setup time.

AC Test Circuit



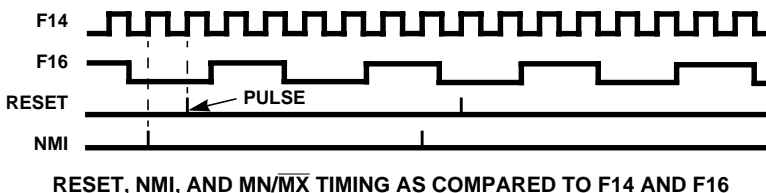
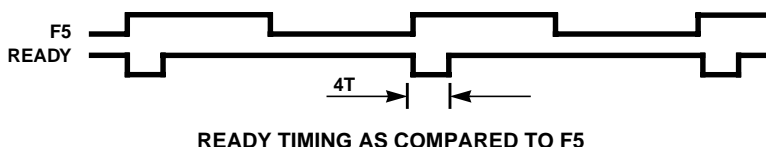
NOTE: Includes stray and jig capacitance.

AC Testing Input, Output Waveform



NOTE: All inputs signals (other than CLK) must switch between  $V_{IL}$  Max -0.4V and  $V_{IH}$  Min +0.4. CLK must switch between 0.4V and  $V_{DD}$  -0.4V. TR and TF must be less than or equal to 15ns. CLK TR and TF must be less than or equal to 10ns.

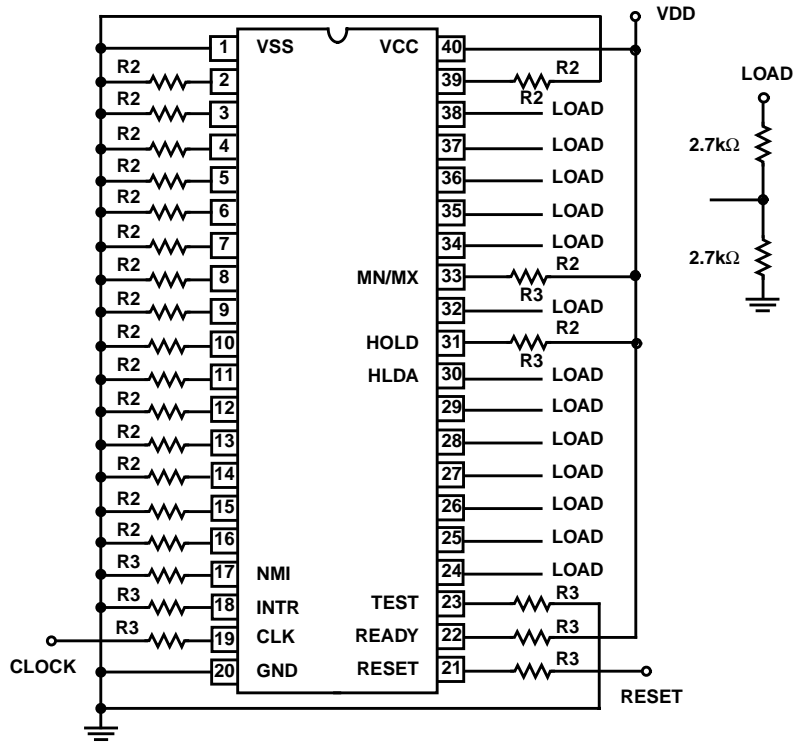
Timing Diagrams



NOTES:

- F0 = 100kHz, 50% duty cycle square wave.  
F1 = F0/2, F2 = F1/2 . . . F16 = F15/2.
- READY, RESET, and NMI timing are as shown: T = 10 $\mu$ s.
- All signals have rise/fall time limits: 100ns < t-rise, t-fall < 500ns.
- RESET has a pulse width = 8T and occurs every two cycles of F16.
- NMI has a pulse width = 4T and occurs every two cycles of F16.
- $MN/\bar{MX}$  is a 50% duty cycle square wave and changes every eight cycles of F16.

**Irradiation Circuit**

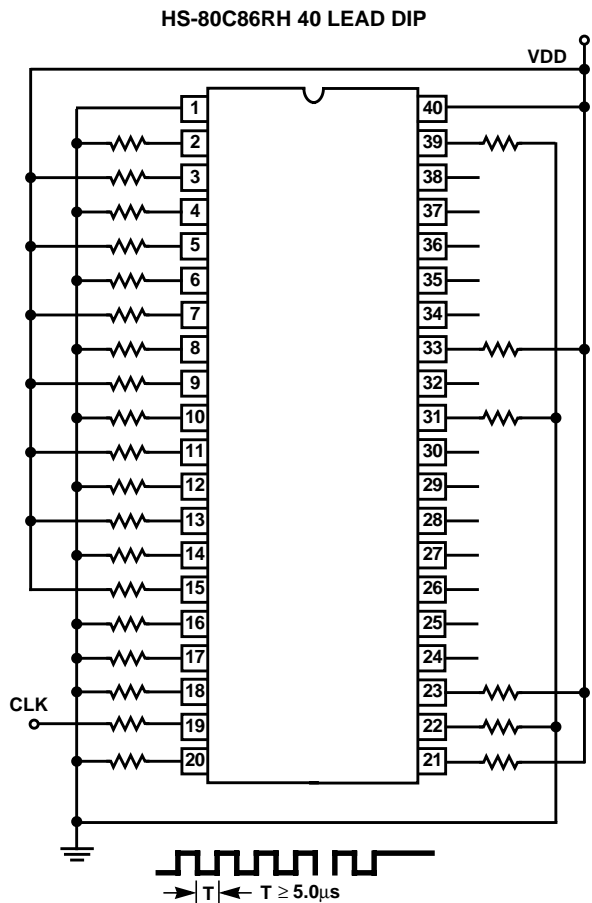


**NOTES:**

10. VDD = 5.0V ±0.5V
11. R2 = 3.3kΩ, R3 = 47kΩ
12. Pins Tied to GND: 1-18, 20, 23, 39  
 Pins Tied to VCC: 22, 31, 33, 40  
 Pins With Loads: 24-29, 30, 32, 34-38  
 Pins Brought Out: 19 (Clock), 21 (Reset)
13. Clock and reset should be brought out separately so they can be toggled before irradiation.
14. Group E Sample Size is 2 Die/Wafer.



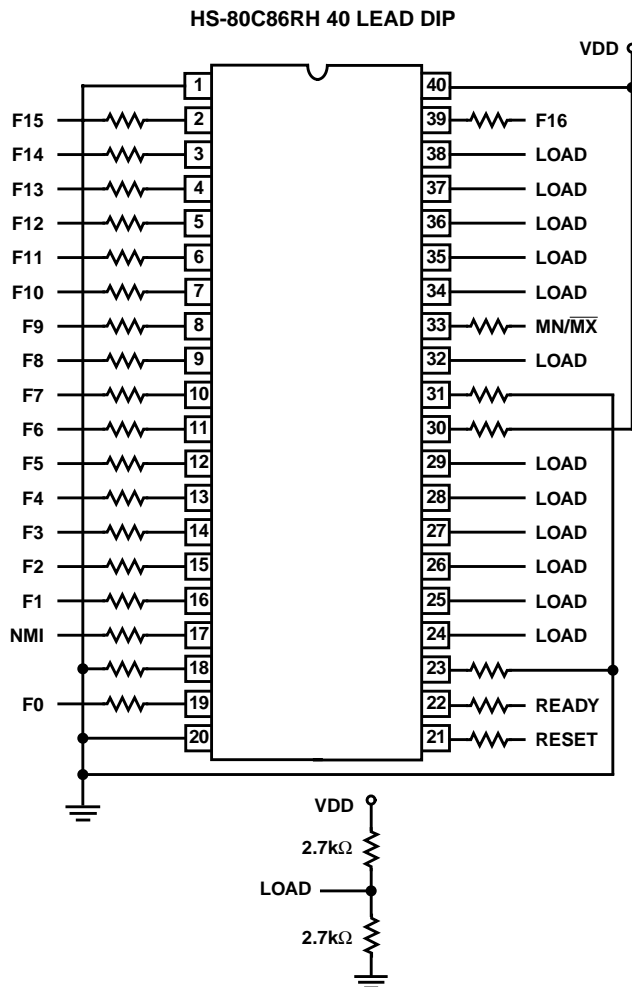
**Burn-In Circuits**



**STATIC**

**NOTES:**

- 15. VDD = +6.5V ±10%.
- 16. T<sub>A</sub> = 125°C Minimum.
- 17. Part is Static Sensitive.
- 18. Voltages Must Be Ramped.
- 19. Package: 40 Lead DIP.
- 20. Resistors:
  - 10kΩ ±10% (Pins 17, 18, 21-23, 31, 33)
  - 2.7kΩ ±5% (Pins 2-16, 39)
  - 1.0kΩ ±5% 1/10W Min (Pin 19)
 Minimum of 5 CLK Pulses  
 After Initial Pulses, CLK is Left High  
 Pulses are 50% Duty Cycle Square Wave



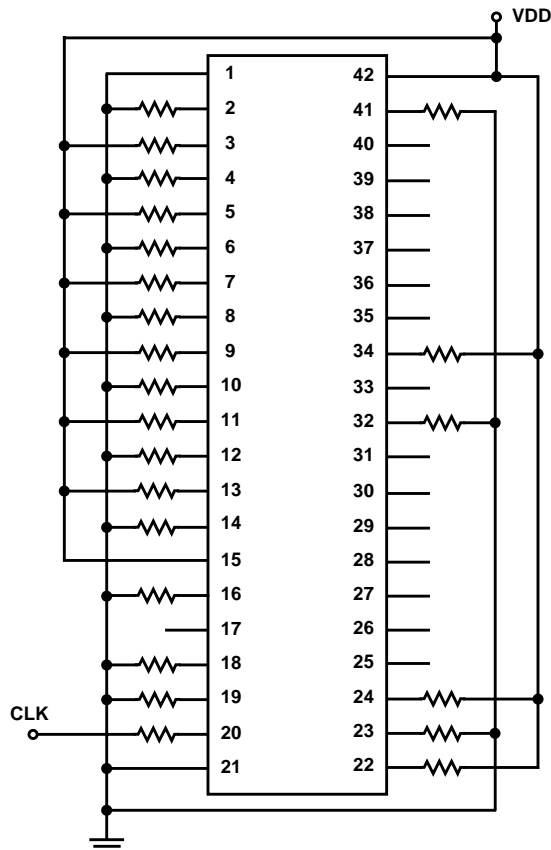
**DYNAMIC**

**NOTES:**

- 21. VDD = 6.5V ±5% (Burn-In).
- 22. VDD = 6.0V ±5% (Life Test).
- 23. T<sub>A</sub> = 125°C.
- 24. Package: 40 Lead DIP.
- 25. Part is Static Sensitive.
- 26. Voltage Must Be Ramped.
- 27. Resistors:
  - 10kΩ (Pins 17, 18, 21, 22, 23, 33)
  - 3.3kΩ (Pins 2-16, 19, 30, 31, 39)
  - 2.7kΩ Loads As Indicated
 All Resistors Are At Least 1/8W, ±10%  
 F0 = 100kHz, F1 = F0/2, F2 = F1/2 . . .  
 RESET, NMI low after initialization.  
 READY pulsed low every 320ms  
 MN/M<sub>X</sub> changes state every 5.24s

**Burn-In Circuits** (Continued)

HS-80C86RH 42 LEAD FLATPACK

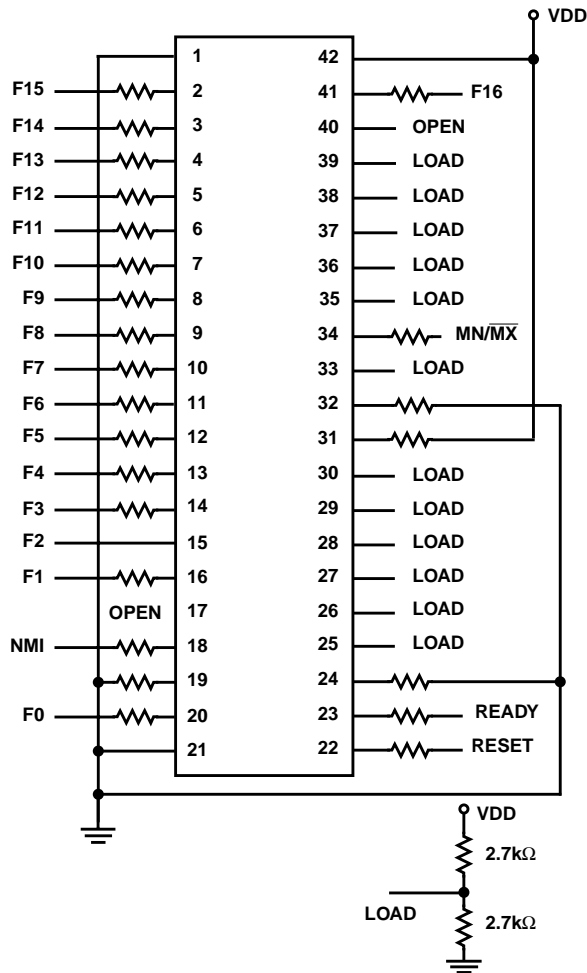


**STATIC**

NOTES:

- 28. VDD = +6.5V ±10%.
- 29. T<sub>A</sub> = 125°C Minimum.
- 30. Part is Static Sensitive.
- 31. Voltages Must Be Ramped.
- 32. Package: 42 Lead Flatpack.
- 33. Resistors:
  - 10kΩ ±10% (Pins 18, 19, 22-24, 32, 34)
  - 2.7kΩ ±5% (Pins 2-16, 41)
  - 1.0kΩ ±5% 1/10W Min (Pin 20)
  - Minimum of 5 CLK Pulses
  - After Initial Pulses, CLK is Left High
  - Pulses are 50% Duty Cycle Square Wave

HS-80C86RH 42 LEAD FLATPACK



**DYNAMIC**

NOTES:

- 34. VDD = 6.5V ±5% (Burn-In).
- 35. VDD = 6.0V ±5% (Life Test).
- 36. T<sub>A</sub> = 125°C.
- 37. Package: 42 Lead Flatpack.
- 38. Part is Static Sensitive.
- 39. Voltage Must Be Ramped.
- 40. Resistors:
  - 10kΩ (Pins 17, 18, 19, 22, 23, 24, 34)
  - 3.3kΩ (Pins 2-16, 20, 31, 32, 41)
  - 2.7kΩ Loads As Indicated
  - All Resistors Are At Least 1/8W, ±10%
  - F0 = 100kHz, F1 = F0/2, F2 = F1/2 . . .
  - RESET, NMI low after initialization.
  - READY pulsed low every 320μs
  - MN/MX changes state every 5.24s

Waveforms

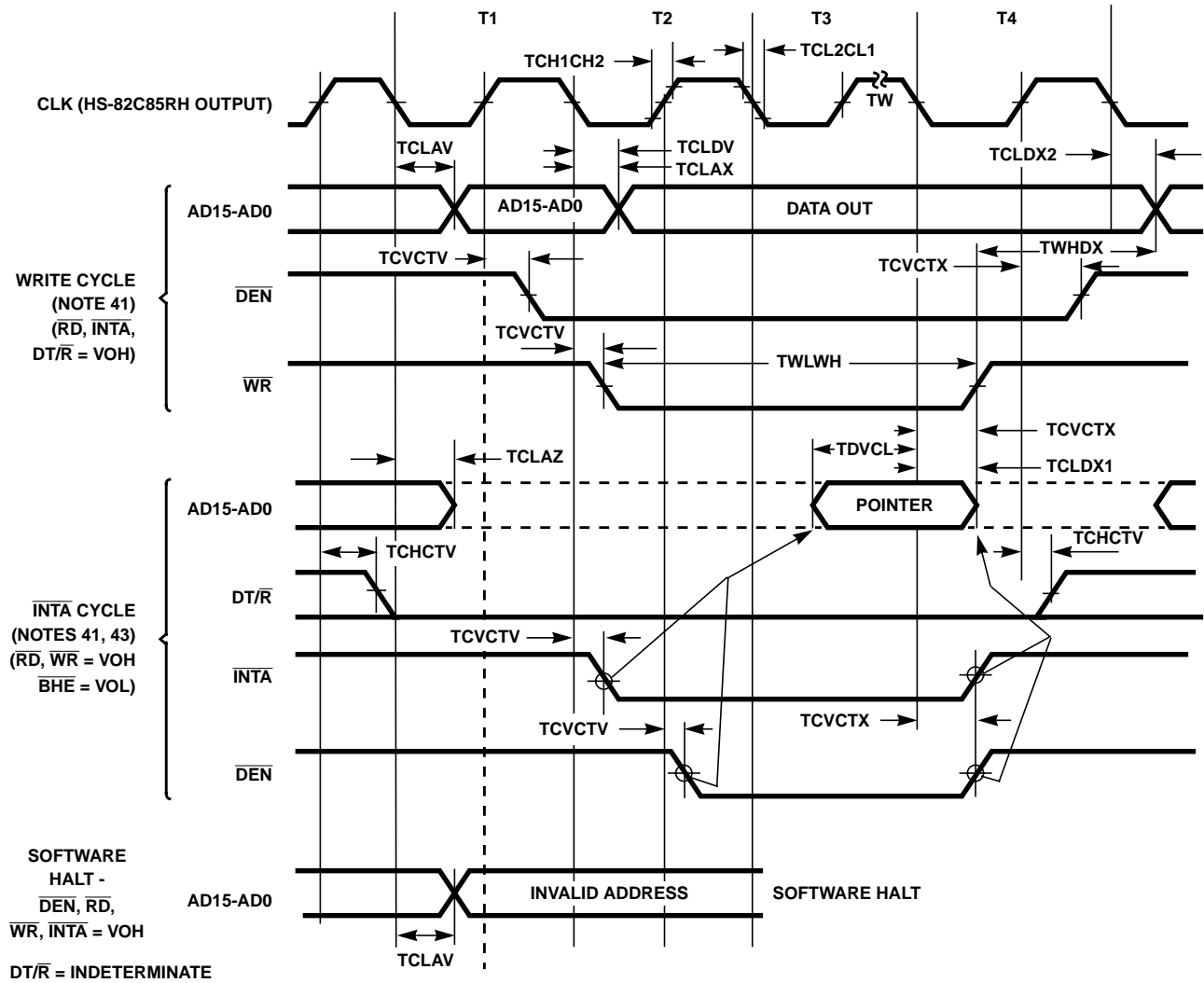


FIGURE 1. BUS TIMING - MINIMUM MODE SYSTEM

NOTES:

41. All signals switch between VOH and VOL unless otherwise specified.
42. RDY is sampled near the end of T2, T3, TW to determine if TW machines states are to be inserted.
43. Two  $\overline{INTA}$  cycles run back-to-back. The HS-80C86RH local ADDR/DATA bus is inactive during both  $\overline{INTA}$  cycles. Control signals are shown for the second  $\overline{INTA}$  cycle.
44. Signals at HS-82C85RH are shown for reference only.
45. All timing measurements are made at 1.5V unless otherwise noted.

Waveforms (Continued)

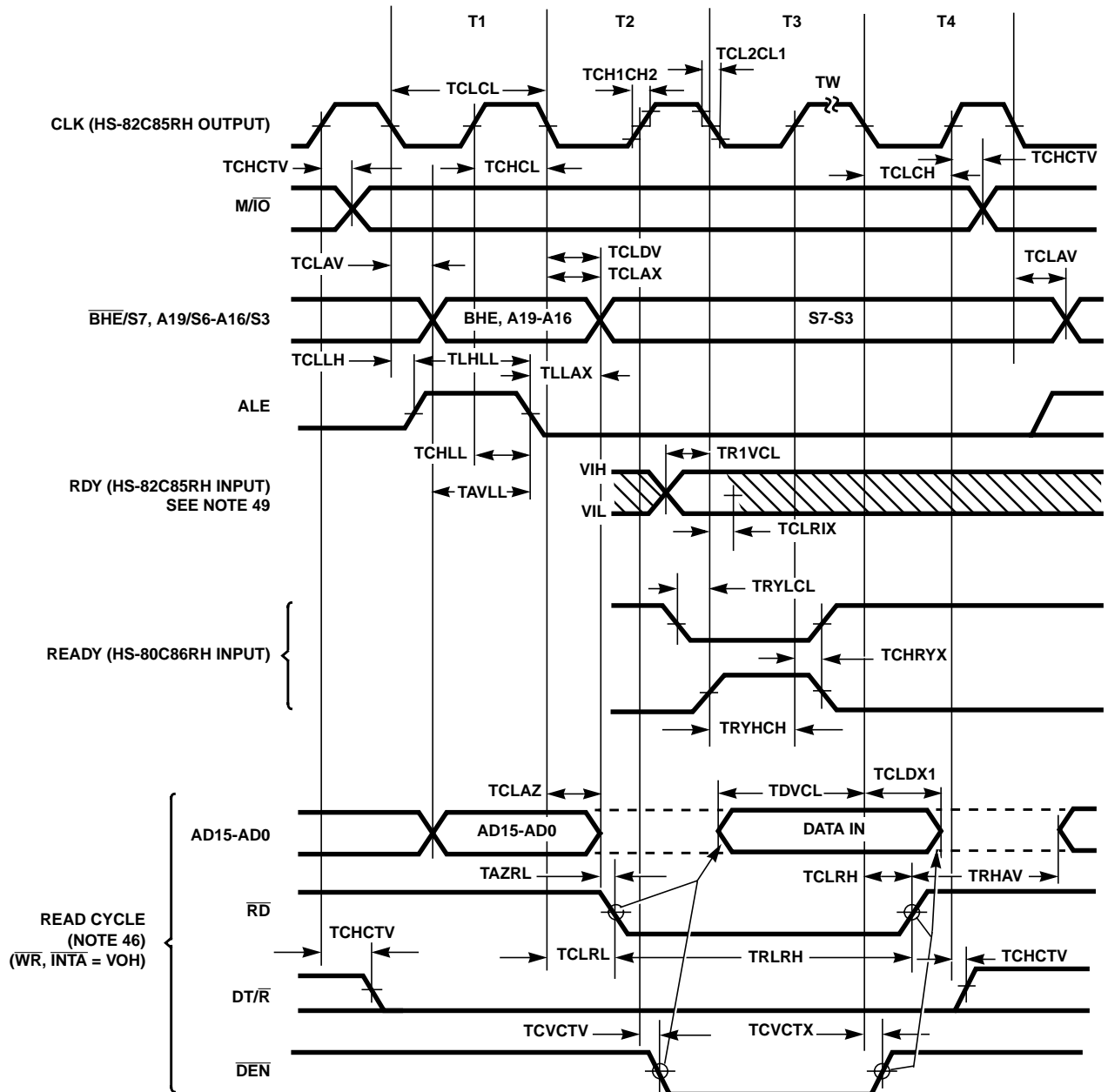


FIGURE 2. BUS TIMING - MINIMUM MODE SYSTEM

NOTES:

- 46. All signals switch between VOH and VOL unless otherwise specified.
- 47. RDY is sampled near the end of T2, T3, TW to determine if TW machines states are to be inserted.
- 48. Two  $\overline{INTA}$  cycles run back-to-back. The HS-80C86RH local ADDR/DATA bus is inactive during both  $\overline{INTA}$  cycles. Control signals are shown for the second  $\overline{INTA}$  cycle.
- 49. Signals at HS-82C85RH are shown for reference only.
- 50. All timing measurements are made at 1.5V unless otherwise noted.

Waveforms (Continued)

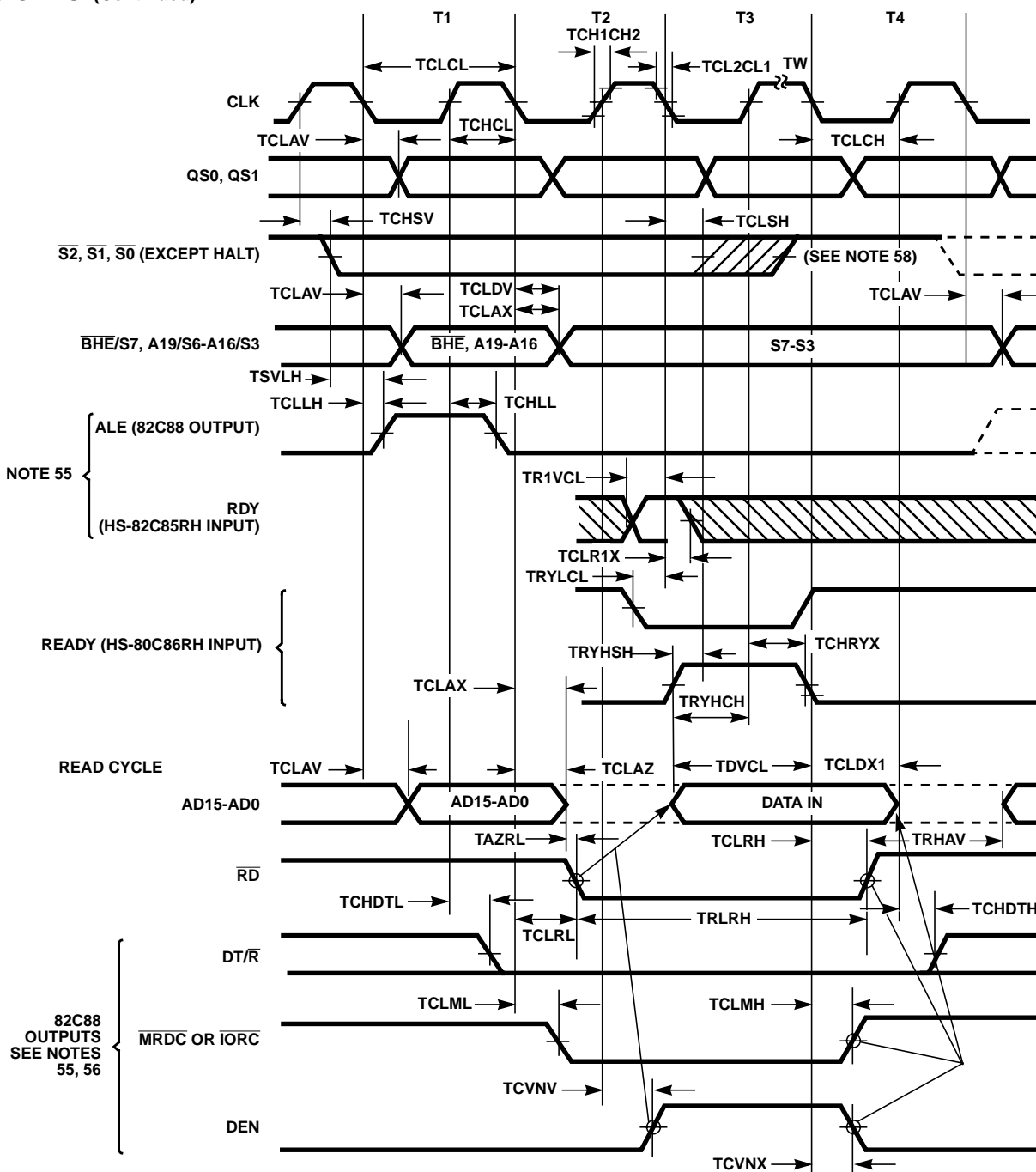


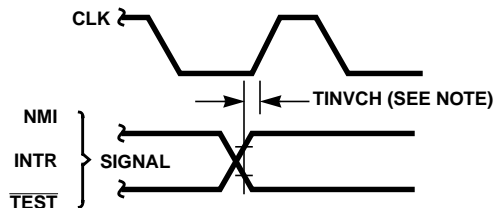
FIGURE 3. BUS TIMING - MAXIMUM MODE SYSTEM

NOTES:

- 51. All signals switch between VOH and VOL unless otherwise specified.
- 52. RDY is sampled near the end of T2, T3, TW to determine if TW machines states are to be inserted.
- 53. Cascade address is valid between first and second  $\overline{INTA}$  cycle.
- 54. Two  $\overline{INTA}$  cycles run back-to-back. The HS-80C86RH local ADDR/DATA bus is inactive during both  $\overline{INTA}$  cycles. Control for pointer address is shown for the second  $\overline{INTA}$  cycle.
- 55. Signals at HS-82C85RH or 82C88 are shown for reference only.
- 56. The issuance of the 82C88 command and control signals ( $\overline{MRDC}$ ,  $\overline{MWTC}$ ,  $\overline{AMWC}$ ,  $\overline{IORC}$ ,  $\overline{IOWC}$ ,  $\overline{AIOWC}$ ,  $\overline{INTA}$  and DEN) lags the active high 82C88 CEN.
- 57. All timing measurements are made at 1.5V unless otherwise noted.
- 58. Status inactive in state just prior to T4.



Waveforms (Continued)



NOTE: Setup Requirements for asynchronous signals only to guarantee recognition at next CLK.

FIGURE 5. ASYNCHRONOUS SIGNAL RECOGNITION

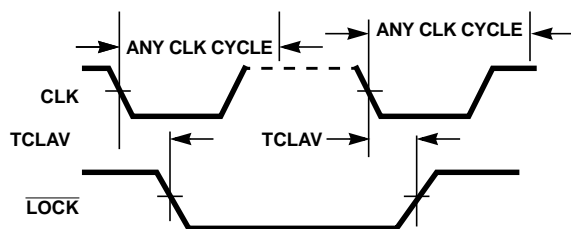
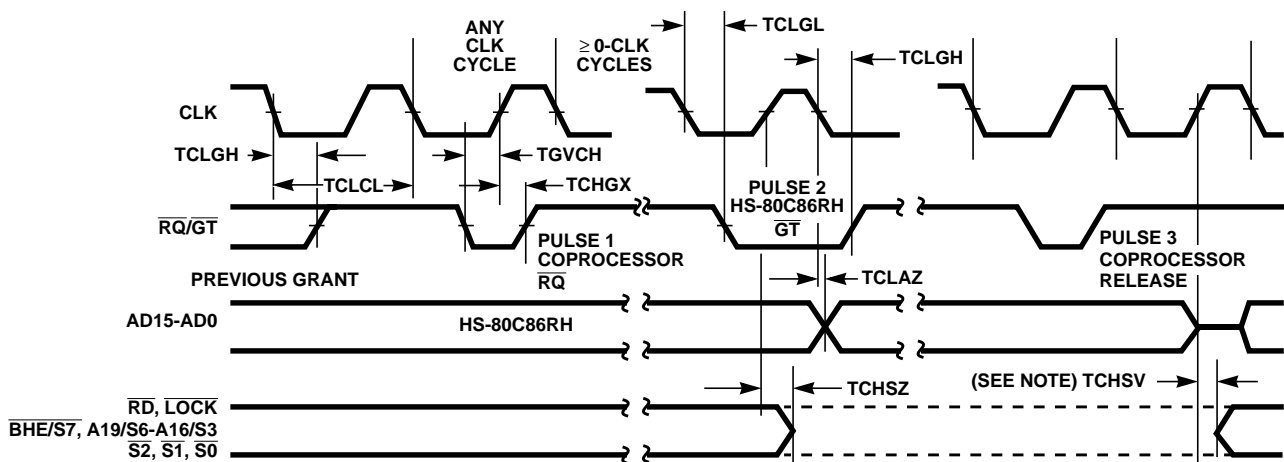


FIGURE 6. BUS LOCK SIGNAL TIMING (MAXIMUM MODE ONLY)



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

FIGURE 7. REQUEST/GRANT SEQUENCE TIMING (MAXIMUM MODE ONLY)

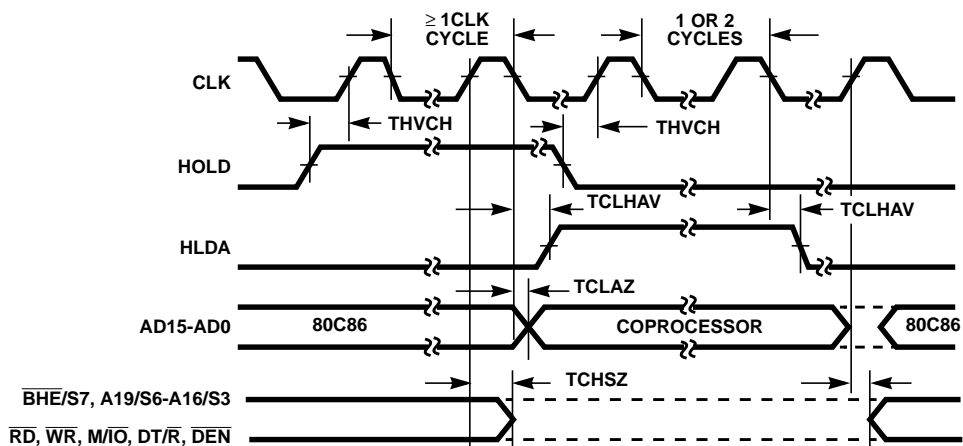


FIGURE 8. HOLD/HOLD ACKNOWLEDGE TIMING (MINIMUM MODE ONLY)

## Functional Description

### Static Operation

All HS-80C86RH 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 CMOS HS-80C86RH can operate from DC to 5MHz. 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 HS-80C86RH 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 HS-80C86RH 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, the HS-80C86RH power requirement is the standby current, (500µA maximum).

### Internal Architecture

The internal functions of the HS-80C86RH processor are partitioned logically into two processing units. The first is the Bus Interface Unit (BIU) and the second is the Execution Unit (EU) as shown in the CPU functional diagram.

These units can interact directly but for the most part perform as separate asynchronous operational processors. The bus interface unit provides the functions related to instruction fetching and queuing, operand fetch and store, and address relocation. This unit also provides the basic bus control. The overlap of instruction pre-fetching provided by this unit serves to increase processor performance through improved bus bandwidth utilization. Up to 6 bytes of the instruction stream can be queued while waiting for decoding and execution.

The instruction stream queuing mechanism allows the BIU to keep the memory utilized very efficiently. Whenever there is space for at least 2 bytes in the queue, the BIU will attempt a word fetch memory cycle. This greatly reduces "dead-time" on the memory bus. The queue acts as a First-In-First-Out (FIFO) buffer, from which the EU extracts instruction bytes as required. If the queue is empty (following a branch instruction, for example), the first byte into the queue immediately becomes available to the EU.

The execution unit receives pre-fetched instructions from the BIU queue and provides un-relocated operand addresses to the BIU. Memory operands are passed through the BIU for processing by the EU, which passes results to the BIU for storage.

## Memory Organization

The processor provides a 20-bit address to memory, which locates the byte being referenced. The memory is organized as a linear array of up to 1 million bytes, addressed as 00000(H) to FFFFF(H). The memory is logically divided into code, data, extra and stack segments of up to 64K bytes each, with each segment falling on 16 byte boundaries. (See Figure 9).

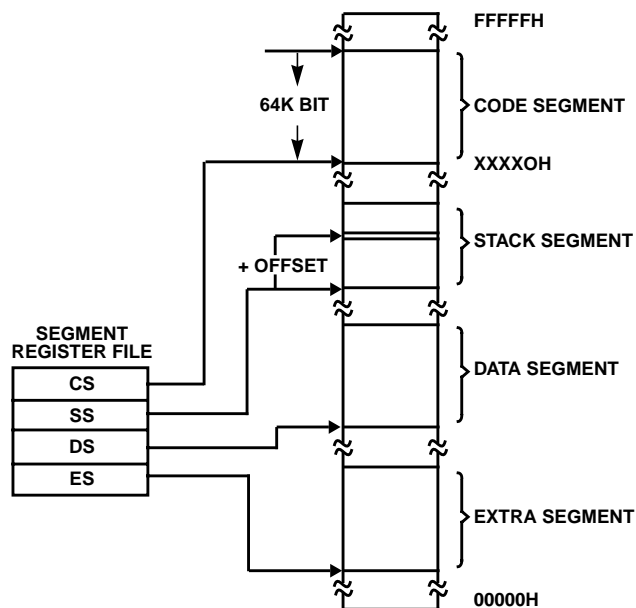


FIGURE 9. HS-80C86RH MEMORY ORGANIZATION

TABLE 1.

TYPE OF MEMORY REFERENCE	DEFAULT SEGMENT BASE	ALTERNATE SEGMENT BASE	OFFSET
Instruction Fetch	CS	None	IP
Stack Operation	SS	None	SP
Variable (Except Following)	DS	CS, ES, SS	Effective Address
String Source	DS	CS, ES, SS	SI
String Destination	ES	None	DI
BP Used as Base Register	SS	CS, DS, ES	Effective Address

All memory references are made relative to base addresses contained in high speed segment registers. The segment types were chosen based on the addressing needs of programs. The segment register to be selected is automatically chosen according to the specific rules of Table 1. All information in one segment type share the same logical attributes (e.g., code or data). By structuring memory



into relocatable areas of similar characteristics and by automatically selecting segment registers, programs are shorter, faster and more structured. (See Table 1).

Word (16-bit) operands can be located on even or odd address boundaries and are thus not constrained to even boundaries as is the case in many 16-bit computers. For address and data operands, the least significant byte of the word is stored in the lower valued address location and the most significant byte in the next higher address location. The BIU automatically performs the proper number of memory accesses, one if the word operand is on an even byte boundary and two if it is on an odd byte boundary. Except for the performance penalty, this double access is transparent to the software. The performance penalty does not occur for instruction fetches; only word operands.

Physically, the memory is organized as a high bank (D15-D6) and a low bank (D7-D0) of 512K bytes addressed in parallel by the processor's address lines.

Byte data with even addresses is transferred on the D7-D0 bus lines while odd addressed byte data (A0 HIGH) is transferred on the D15-D6 bus lines. The processor provides two enable signals,  $\overline{BHE}$  and A0, to selectively allow reading from or writing into either an odd byte location, even byte location, or both. The instruction stream is fetched from memory as words and is addressed internally by the processor at the byte level as necessary.

In referencing word data, the BIU requires one or two memory cycles depending on whether the starting byte of the word is on an even or odd address, respectively. Consequently, in referencing word operands performance can be optimized by locating data on even address boundaries. This is an especially useful technique for using the stack, since odd address references to the stack may adversely affect the context switching time for interrupt processing or task multiplexing.

Certain locations in memory are reserved for specific CPU operations (See Figure 10). Locations from address FFFF0H through FFFFH are reserved for operations including a jump to the initial program loading routine. Following RESET, the CPU will always begin execution at location FFFF0H where the jump must be located. Locations 00000H through 003FFH are reserved for interrupt operations. Each of the 256 possible interrupt service routines is accessed through its own pair of 16-bit pointers - segment address pointer and offset address pointer. The first pointer, used as the offset address, is loaded into the 1P and the second pointer, which designates the base address is loaded into the CS. At this point program control is transferred to the interrupt routine. The pointer elements are assumed to have been stored at the respective places in reserved memory prior to occurrence of interrupts.

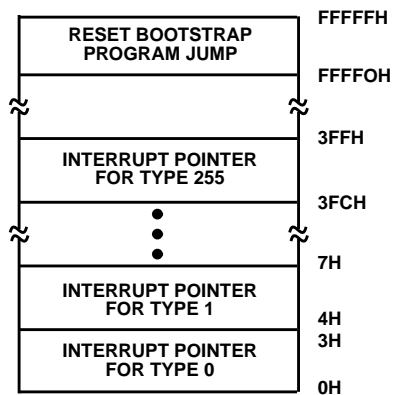


FIGURE 10. RESERVED MEMORY LOCATIONS

**Minimum and Maximum Operation Modes**

The requirements for supporting minimum and maximum HS-80C86RH systems are sufficiently different that they cannot be met efficiently using 40 uniquely defined pins. Consequently, the HS-80C86RH is equipped with a strap pin (MN/MX) which defines the system configuration. The definition of a certain subset of the pins changes, dependent on the condition of the strap pin. When the MN/MX pin is strapped to GND, the HS-80C86RH defines pins 24 through 31 and 34 in maximum mode. When the MN/MX pin is strapped to VDD, the HS-80C86RH generates bus control signals itself on pins 24 through 31 and 34.

**Bus Operation**

The HS-80C86RH has a combined address and data bus commonly referred to as a time multiplexed bus. This technique provides the most efficient use of pins on the processor while permitting the use of a standard 40-lead package. This "local bus" can be buffered directly and used throughout the system with address latching provided on memory and I/O modules. In addition, the bus can also be demultiplexed at the processor with a single set of 82C82 latches if a standard non-multiplexed bus is desired for the system.

Each processor bus cycle consists of at least four CLK cycles. These are referred to as T1, T2, T3 and T4 (see Figure 11). The address is emitted from the processor during T1 and data transfer occurs on the bus during T3 and T4. T2 is used primarily for changing the direction of the bus during read operations. In the event that a "NOT READY" indication is given by the addressed device, "Wait" states (TW) are inserted between T3 and T4. Each inserted wait state is the same duration as a CLK cycle. Idle periods occur between HS-80C86RH driven bus cycles whenever the processor performs internal processing.

During T1 of any bus cycle, the ALE (Address Latch Enable) signal is emitted (by either the processor or the 82C88 bus controller, depending on the MN/MX strap). At the trailing edge of this pulse, a valid address and certain status information for the cycle may be latched.

Status bits  $\overline{S0}$ ,  $\overline{S1}$  and  $\overline{S2}$  are used by the bus controller, in maximum mode, to identify the type of bus transaction according to Table 2.

TABLE 2.

$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	CHARACTERISTICS
0	0	0	Interrupt Acknowledge
0	0	1	Read I/O Port
0	1	0	Write I/O Port
0	1	1	Halt
1	0	0	Instruction Fetch
1	0	1	Read Data from Memory
1	1	0	Write Data to Memory
1	1	1	Passive (no bus cycle)

Status bits S3 through S7 are time multiplexed with high order address bits and the  $\overline{BHE}$  signal, and are therefore valid during T2 through T4. S3 and S4 indicate which segment register (see Instruction Set Description) was used for this bus cycle in forming the address, according to Table 3.

TABLE 3.

S4	S3	CHARACTERISTICS
0 (Low)	0	Alternate Data (extra segment)
0	1	Stack
1 (High)	0	Code or None
1	1	Data

S5 is a reflection of the PSW interrupt enable bit. S6 is always zero and S7 is a spare status bit.

**I/O Addressing**

In the HS-80C86RH, I/O operations can address up to a maximum of 64K I/O byte registers or 32K I/O word registers. The I/O address appears in the same format as the memory address on bus lines A15-A0. The address lines A19-A16 are zero in I/O operations. The variable I/O instructions which use register DX as a pointer have full address capability while the direct I/O instructions directly address one or two of the 256 I/O byte locations in page 0 of the I/O address space.

I/O ports are addressed in the same manner as memory locations. Even addressed bytes are transferred on the D7-D0 bus lines and odd addressed bytes on D15-D8. Care must be taken to ensure that each register within an 8-bit peripheral located on the lower portion of the bus be addressed as even.

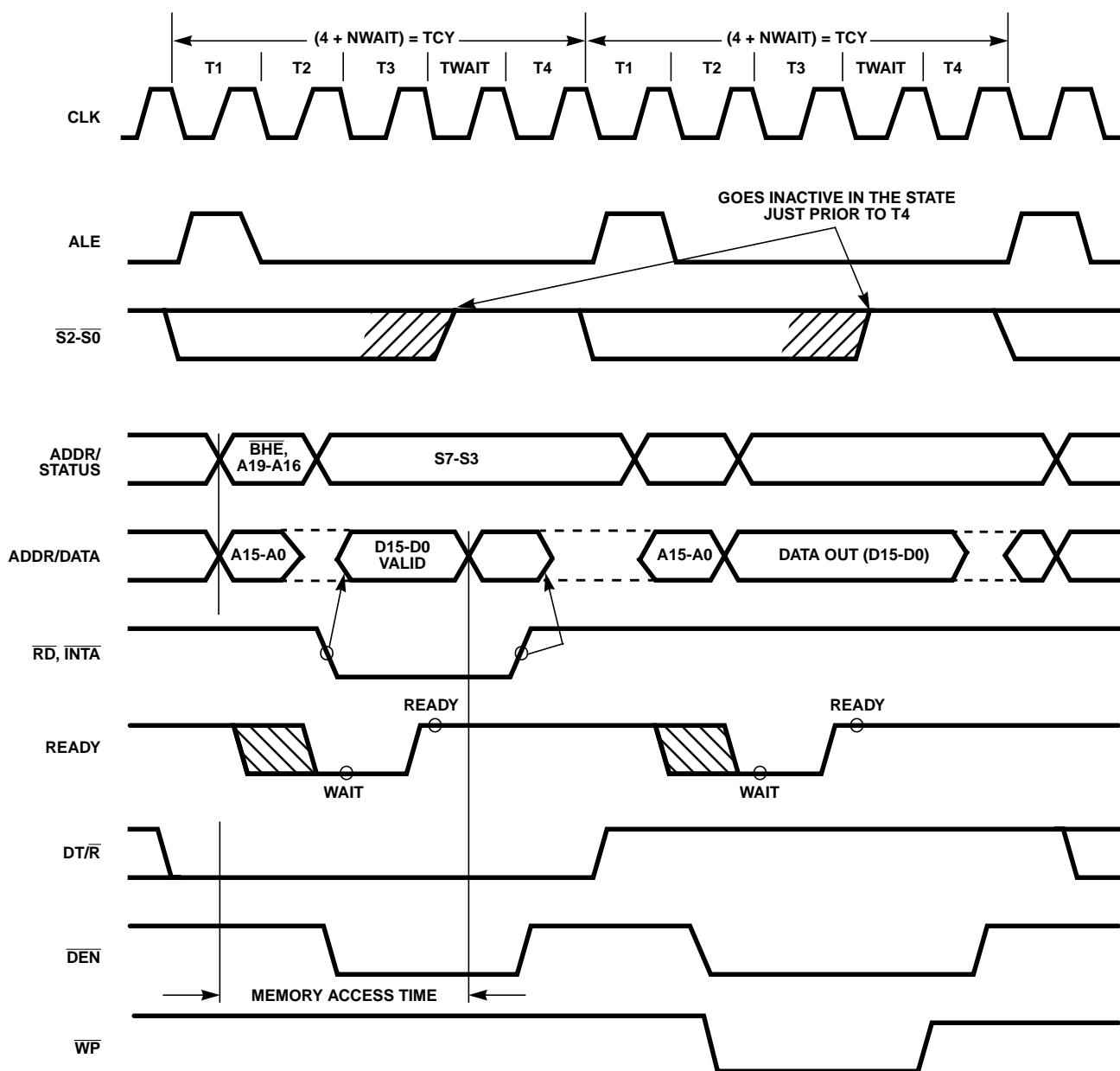


FIGURE 11. BASIC SYSTEM TIMING

## External Interface

### Processor RESET and Initialization

Processor initialization or start up is accomplished with activation (HIGH) of the RESET pin. The HS-80C86RH RESET is required to be HIGH for greater than 4 CLK cycles. The HS-80C86RH will terminate operations on the high-going edge of RESET and will remain dormant as long as RESET is HIGH. The low-going transition of RESET triggers an internal reset sequence for approximately 7 CLK cycles. After this interval, the HS-80C86RH operates normally beginning with the instruction in absolute location FFFFOH. (See Figure 10). The RESET input is internally synchronized to the processor clock. At initialization, the

HIGH-to-LOW transition of RESET must occur no sooner than 50 $\mu$ s (or 4 CLK cycles, whichever is greater) after power-up, to allow complete initialization of the HS-80C86RH.

NMI will not be recognized prior to the second clock cycle following the end of RESET. If NMI is asserted sooner than 9 CLK cycles after the end of RESET, the processor may execute one instruction before responding to the interrupt.

### Bus Hold Circuitry

To avoid high current conditions caused by floating inputs to CMOS devices and to eliminate need for pull-up/down resistors, "bus-hold" circuitry has been used on the HS-80C86RH pins 2-16, 26-32 and 34-39. (See Figures 12A

and 12B). 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 400µA 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.

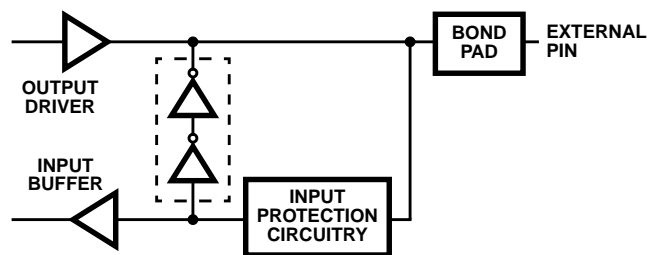


FIGURE 12A. BUS HOLD CIRCUITRY PIN 2-16, 34-39

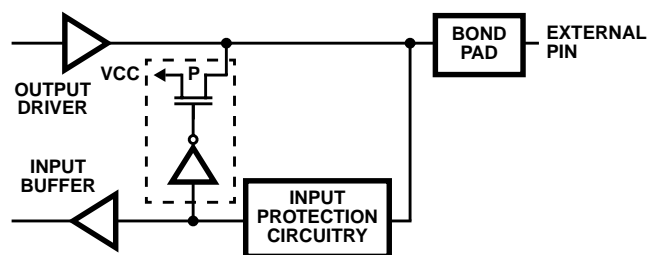


FIGURE 12B. BUS HOLD CIRCUITRY PIN 26-32

### Interrupt Operations

Interrupt operations fall into two classes: software or hardware initiated. The software initiated interrupts and software aspects of hardware interrupts are specified in the Instruction Set Description. Hardware interrupts can be classified as non-maskable or maskable.

Interrupts result in a transfer of control to a new program location. A 256-element table containing address pointers to the interrupt service routine locations resides in absolute locations 0 through 3FFH, which are reserved for this purpose. Each element in the table is 4 bytes in size and corresponds to an interrupt “type”. An interrupting device supplies an 8-bit type number during the interrupt acknowledge sequence, which is used to “vector” through the appropriate element to the interrupt service routine location. All flags and both the Code Segment and Instruction Pointer register are saved as part of the  $\overline{\text{INTA}}$  sequence. These are restored upon execution of an Interrupt Return (IRET) instruction.

### Non-Maskable Interrupt (NMI)

The processor provides a single non-maskable interrupt pin (NMI) which has higher priority than the maskable interrupt request pin (INTR). A typical use would be to activate a power failure routine. The NMI is edge-triggered on a LOW-to-HIGH transition. The activation of this pin causes a type 2 interrupt.

NMI is required to have a duration in the HIGH state of greater than 2 CLK cycles, but is not required to be synchronized to the clock. Any positive transition of NMI is latched on-chip and will be serviced at the end of the current instruction or between whole moves of a block-type instruction. Worst case response to NMI would be for multiply, divide, and variable shift instructions. There is no specification on the occurrence of the low-going edge; it may occur before, during or after the servicing of NMI. Another positive edge triggers another response if it occurs after the start of the NMI procedure. The signal must be free of logical spikes in general and be free of bounces on the low-going edge to avoid triggering extraneous responses.

### Maskable Interrupt (INTR)

The HS-80C86RH provides a single interrupt request input (INTR) which can be masked internally by software with the resetting of the interrupt enable flag (IF) status bit. The interrupt request signal is level triggered. It is internally synchronized during each clock cycle on the high-going edge of CLK. To be responded to, INTR must be present (HIGH) during the clock period preceding the end of the current instruction or the end of a whole move for a block-type instruction. INTR may be removed anytime after the falling edge of the first  $\overline{\text{INTA}}$  signal. During the interrupt response sequence further interrupts are disabled. The enable bit is reset as part of the response to any interrupt (INTR, NMI, software interrupt or single-step), although the FLAGS register which is automatically pushed onto the stack reflects the state of the processor prior to the interrupt. Until the old FLAGS register is restored the enable bit will be zero unless specifically set by an instruction.

During the response sequence (Figure 13) the processor executes two successive (back-to-back) interrupt acknowledge cycles. The HS-80C86RH emits the  $\overline{\text{LOCK}}$  signal (Max mode only) from T2 of the first bus cycle until T2 of the second. A local bus “hold” request will not be honored until the end of the second bus cycle. In the second bus cycle, a byte is supplied to the HS-80C86RH by the HS-82C89ARH Interrupt Controller, which identifies the source (type) of the interrupt. This byte is multiplied by four and used as a pointer into the interrupt vector lookup table. An INTR signal left HIGH will be continually responded to within the limitations of the enable bit and sample period. The INTERRUPT RETURN instruction includes a FLAGS pop which returns the status of the original interrupt enable bit when it restores the FLAGS.

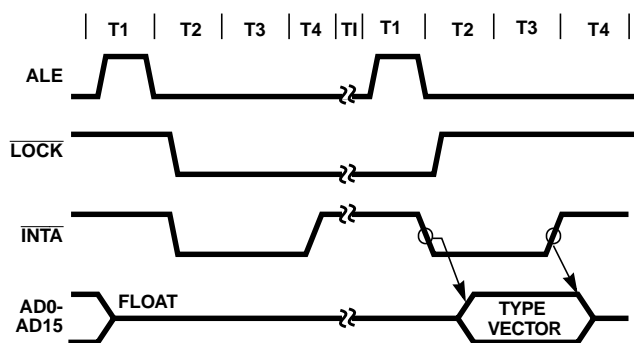


FIGURE 13. INTERRUPT ACKNOWLEDGE SEQUENCE

**Halt**

When a software “HALT” instruction is executed the processor indicates that it is entering the “HALT” state in one of two ways depending upon which mode is strapped. In minimum mode, the processor issues one ALE with no qualifying bus control signals. In maximum mode the processor issues appropriate HALT status on  $\overline{S2}$ ,  $\overline{S1}$ ,  $\overline{S0}$  and the 82C88 bus controller issues one ALE. The HS-80C86RH will not leave the “HALT” state when a local bus “hold” is entered while in “HALT”. In this case, the processor reissues the HALT indicator at the end of the local bus hold. An NMI or interrupt request (when interrupts enabled) or RESET will force the HS-80C86RH out of the “HALT” state.

**Read/Modify/Write (Semaphore)**

**Operations Via Lock**

The  $\overline{LOCK}$  status information is provided by the processor when consecutive bus cycles are required during the execution of an instruction. This gives the processor the capability of performing read/modify/write operations on memory (via the Exchange Register With Memory instruction, for example) without another system bus master receiving intervening memory cycles. This is useful in multiprocessor system configurations to accomplish “test and set lock” operations. The  $\overline{LOCK}$  signal is activated (forced LOW) in the clock cycle following decoding of the software “LOCK” prefix instruction. It is deactivated at the end of the last bus cycle of the instruction following the “LOCK” prefix instruction. While  $\overline{LOCK}$  is active a request on a  $\overline{RQ}/\overline{GT}$  pin will be recorded and then honored at the end of the  $\overline{LOCK}$ .

**External Synchronization Via TEST**

As an alternative to interrupts, the HS-80C86RH provides a single software-testable input pin (TEST). This input is utilized by executing a WAIT instruction. The single WAIT instruction is repeatedly executed until the TEST input goes active (LOW). The execution of WAIT does not consume bus cycles once the queue is full.

If a local bus request occurs during WAIT execution, the HS-80C86RH three-states all output drivers while inputs and I/O pins are held at valid logic levels by internal bus-hold circuits. If interrupts are enabled, the HS-80C86RH will recognize interrupts and process them when it regains control of the bus. The WAIT instruction is then refetched, and reexecuted.

**Basic System Timing**

Typical system configurations for the processor operating in minimum mode and in maximum mode are shown in Figures 14A and 14B, respectively. In minimum mode, the MN/ $\overline{MX}$  pin is strapped to VDD and the processor emits bus control signals (e.g.  $\overline{RD}$ ,  $\overline{WR}$ , etc.) directly. In maximum mode, the MN/ $\overline{MX}$  pin is strapped to GND and the processor emits coded status information which the 82C88 bus controller used to generate Multibus™ compatible bus control signals. Figure 11 shows the signal timing relationships.

TABLE 4. HS-80C86RH REGISTER MODEL

AX	AH	AL	ACCUMULATOR
BX	BH	BL	BASE
CX	CH	CL	COUNT
DX	DH	DL	DATA
	SP		STACK POINTER
	BP		BASE POINTER
	SI		SOURCE INDEX
	DI		DESTINATION INDEX
	IP		INSTRUCTION POINTER
	FLAGSH	FLAGSL	STATUS FLAGS
	CS		CODE SEGMENT
	DS		DATA SEGMENT
	SS		STACK SEGMENT
	ES		EXTRA SEGMENT

**System Timing - Minimum System**

The read cycle begins in T1 with the assertion of the Address Latch Enable (ALE) signal. The trailing (low-going) edge of this signal is used to latch the address information, which is valid on the address/data bus (AD0-AD15) at this time, into the 82C82 latches. The  $\overline{BHE}$  and A0 signals address the low, high or both bytes. From T1 to T4 the  $\overline{M/\overline{IO}}$  signal indicates a memory or I/O operation. At T2, the address is removed from the address/data bus and the bus is held at the last valid logic state by internal bus hold devices. The read control signal is also asserted at T2. The read ( $\overline{RD}$ ) signal causes the addressed device to enable its data bus drivers to the local bus. Some time later, valid data will be available on the bus and the addressed device will drive the READY line HIGH. When the processor returns the read signal to a HIGH level, the addressed device will three-

state its bus drivers. If a transceiver is required to buffer the HS-80C86RH local bus, signals DT/R and DEN are provided by the HS-80C86RH.

A write cycle also begins with the assertion of ALE and the emission of the address. The M/ $\overline{IO}$  signal is again asserted to indicate a memory or I/O write operation. In T<sub>2</sub>, immediately following the address emission, the processor emits the data to be written into the addressed location. This data remains valid until at least the middle of T<sub>4</sub>. During T<sub>2</sub>, T<sub>3</sub> and T<sub>W</sub>, the processor asserts the write control signal. The write ( $\overline{WR}$ ) signal becomes active at the beginning of T<sub>2</sub> as opposed to the read which is delayed somewhat into T<sub>2</sub> to provide time for output drivers to become inactive.

The BHE and A<sub>0</sub> signals are used to select the proper byte(s) of the memory/I/O word to be read or written according to Table 5.

TABLE 5.

BHE	A <sub>0</sub>	CHARACTERISTICS
0	0	Whole word
0	1	Upper byte from/to odd address
1	0	Lower byte from/to even address
1	1	None

I/O ports are addressed in the same manner as memory location. Even addressed bytes are transferred on the D<sub>7</sub>-D<sub>0</sub> bus lines and odd address bytes on D<sub>15</sub>-D<sub>6</sub>.

The basic difference between the interrupt acknowledge cycle and a read cycle is that the interrupt acknowledge signal ( $\overline{INTA}$ ) is asserted in place of the read ( $\overline{RD}$ ) signal and the address bus is held at the last valid logic state by internal bus hold devices. (See Figures 12A, 12B). In the second of two successive  $\overline{INTA}$  cycles a byte of information is read from the data bus (D<sub>7</sub>-D<sub>0</sub>) as supplied by the interrupt system logic (i.e., HS-82CS9ARH Priority Interrupt Controller). This byte identifies the source (type) of the interrupt. It is multiplied by four and used as a pointer into an interrupt vector lookup table, as described earlier.

**Bus Timing - Medium and Large Size Systems**

For medium complexity systems the MN/ $\overline{MX}$  pin is connected to GND and the 82C88 Bus Controller is added to the system as well as three 82C82 latches for latching the system address, and a transceiver to allow for bus loading greater than the HS-80C86RH is capable of handling. Bus control signals are generated by the 82C88 instead of the processor in this configuration, although their timing remains relatively the same. The HS-80C86RH status outputs ( $\overline{S_2}$ ,  $\overline{S_1}$ , and  $\overline{S_0}$ ) provide type-of-cycle information and become 82C88 inputs. This bus cycle information specifies read (code, data or I/O), write (data or I/O), interrupt acknowledge, or software halt. The 82C88 issues control signals specifying memory read or write, I/O read or write, or interrupt acknowledge. The 82C88 provides two types of write strobes, normal and advanced, to be applied as required. The normal write strobes have data valid at the leading edge of write. The advanced write strobes have the same timing as read strobes, and hence, data is not valid at the leading edge of write. The transceiver receives the usual T and 0E inputs from the 82C88 DT/ $\overline{R}$  and DEN signals.

For large multiple processor systems, the 82C89 bus arbiter must be added to the system to provide system bus management. In this case, the pointer into the interrupt vector table, which is passed during the second  $\overline{INTA}$  cycle, can be derived from an HS-82C59ARH located on either the local bus or the system bus. The processor's  $\overline{INTA}$  output should drive the SYSB/ $\overline{RESB}$  input of the 82C89 to the proper state when reading the interrupt vector number from the HS-82C59ARH during the interrupt acknowledge sequence and software "poll".

**A Note on Radiation Hardened Product Availability**

There are no immediate plans to develop the 82C88 Bus Controller or the 82C89 Arbiter as radiation hardened integrated circuits.

**A Note on SEU Capability of the HS-80C86RH**

Previous heavy ion testing of the HS-80C86RH has indicated that the SEU threshold of this part is about 6MEV/mg/cm<sup>2</sup>. Based upon these results and other analysis, a deep space galactic cosmic-ray environment will result in an SEU rate of about 0.08 upsets/day.

# HS-80C86RH

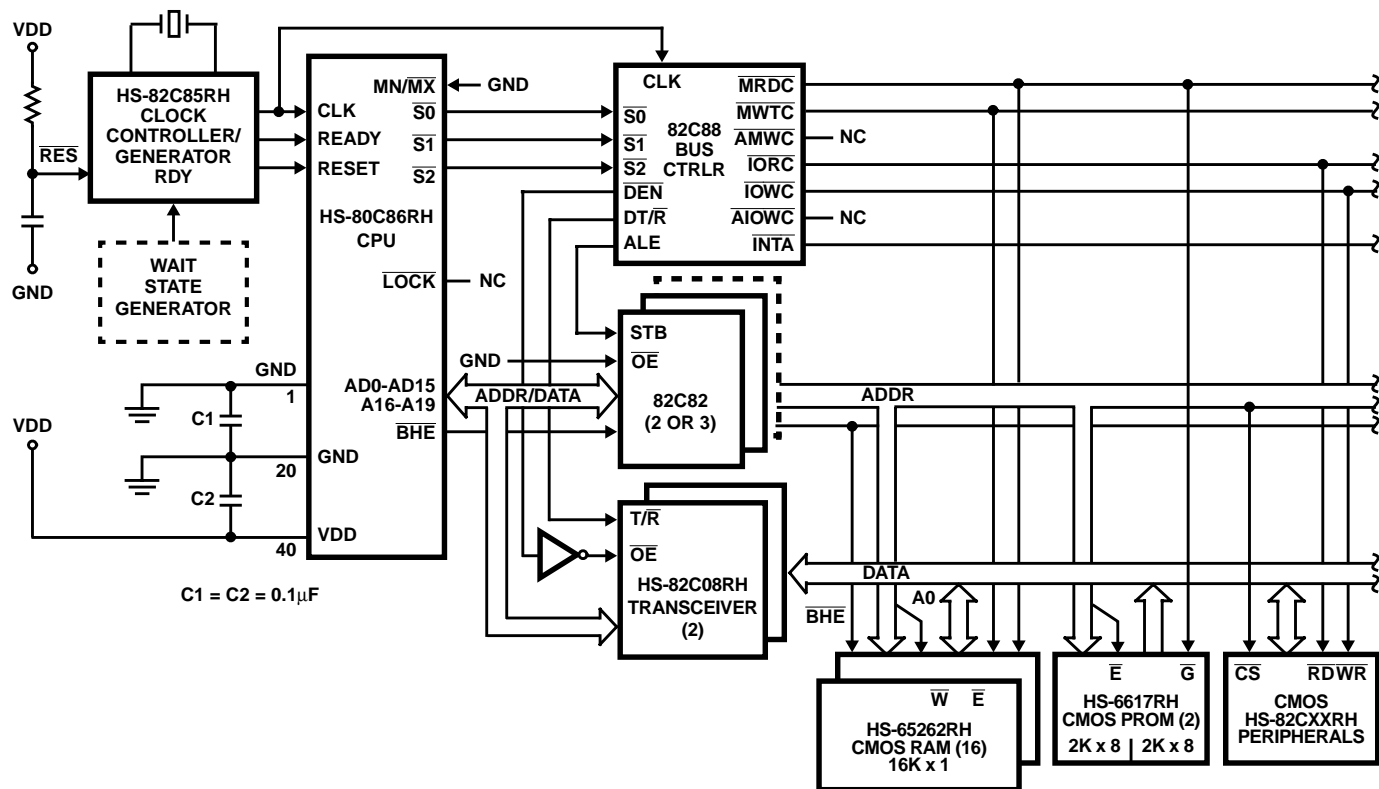


FIGURE 14A. MAXIMUM MODE HS-80C86RH TYPICAL CONFIGURATION

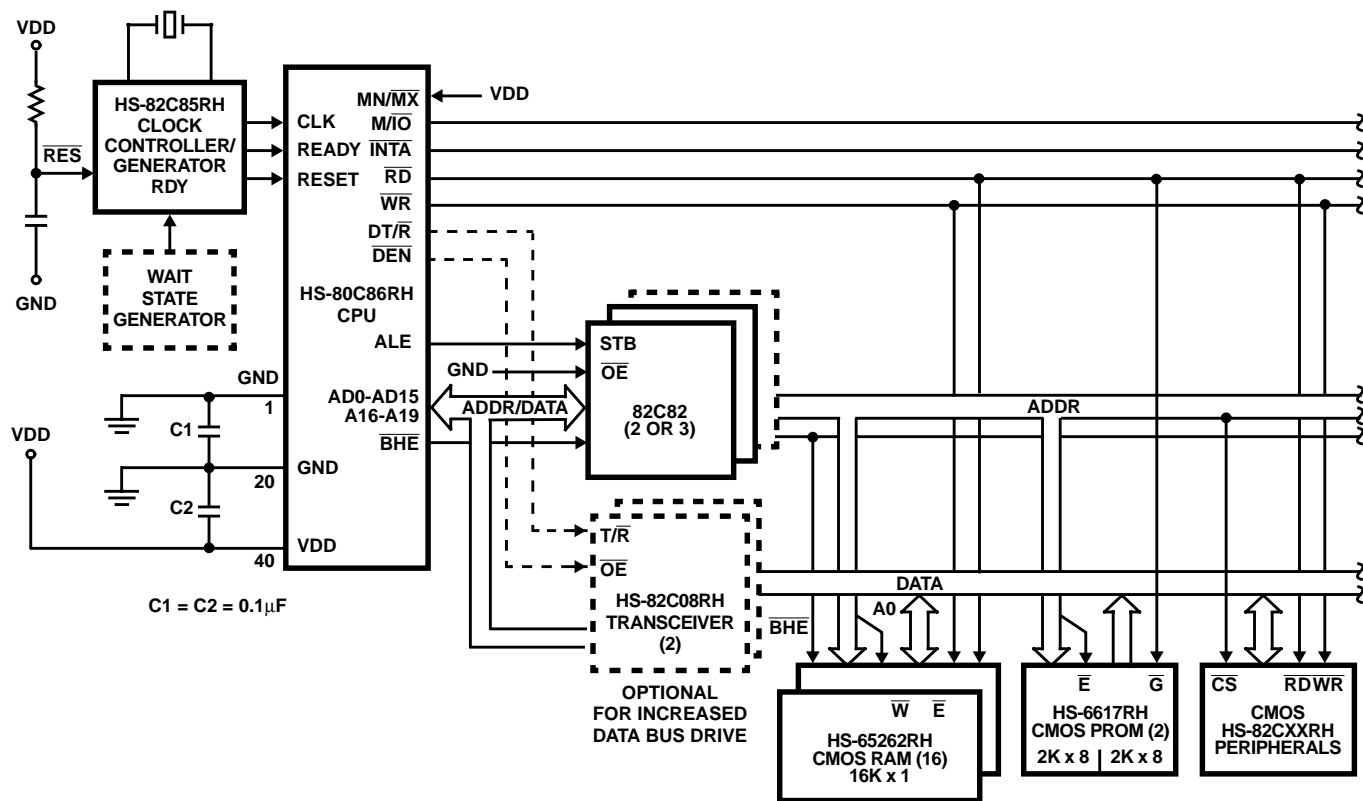


FIGURE 14B. MINIMUM MODE HS-80C86RH TYPICAL CONFIGURATION

**Instruction Set Summary**

MNEMONIC AND DESCRIPTION	INSTRUCTION CODE			
	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
<b>DATA TRANSFER</b>				
<b>MOV = MOVE:</b>				
Register/Memory to/from Register	1 0 0 0 1 0 d w	mod reg r/m		
Immediate to Register/Memory	1 1 0 0 0 1 1 w	mod 0 0 0 r/m	data	data if w 1
Immediate to Register	1 0 1 1 w reg	data	data if w 1	
Memory to Accumulator	1 0 1 0 0 0 w	addr-low	addr-high	
Accumulator to Memory	1 0 1 0 0 0 1 w	addr-low	addr-high	
Register/Memory to Segment Register ††	1 0 0 0 1 1 1 0	mod 0 reg r/m		
Segment Register to Register/Memory	1 0 0 0 1 1 0 0	mod 0 reg r/m		
<b>PUSH = Push:</b>				
Register/Memory	1 1 1 1 1 1 1 1	mod 1 1 0 r/m		
Register	0 1 0 1 0 reg			
Segment Register	0 0 0 reg 1 1 0			
<b>POP = Pop:</b>				
Register/Memory	1 0 0 0 1 1 1 1	mod 0 0 0 r/m		
Register	0 1 0 1 1 reg			
Segment Register	0 0 0 reg 1 1 1			
<b>XCHG = Exchange:</b>				
Register/Memory with Register	1 0 0 0 0 1 1 w	mod reg r/m		
Register with Accumulator	1 0 0 1 0 reg			
<b>IN = Input from:</b>				
Fixed Port	1 1 1 0 0 1 0 w	port		
Variable Port	1 1 1 0 1 1 0 w			
<b>OUT = Output to:</b>				
Fixed Port	1 1 1 0 0 1 1 w	port		
Variable Port	1 1 1 0 1 1 1 w			
<b>XLAT = Translate Byte to AL</b>	1 1 0 1 0 1 1 1			
<b>LEA = Load EA to Register 2</b>	1 0 0 0 1 1 0 1	mod reg r/m		
<b>LDS = Load Pointer to DS</b>	1 1 0 0 0 1 0 1	mod reg r/m		
<b>LES = Load Pointer to ES</b>	1 1 0 0 0 1 0 0	mod reg r/m		
<b>LAHF = Load AH with Flags</b>	1 0 0 1 1 1 1 1			
<b>SAHF = Store AH into Flags</b>	1 0 0 1 1 1 1 0			
<b>PUSHF = Push Flags</b>	1 0 0 1 1 1 0 0			
<b>POPF = Pop Flags</b>	1 0 0 1 1 1 0 1			



**Instruction Set Summary** (Continued)

MNEMONIC AND DESCRIPTION	INSTRUCTION CODE			
	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
<b>ARITHMETIC</b>				
<b>ADD = Add:</b>				
Register/Memory with Register to Either	0 0 0 0 0 d w	mod reg r/m		
Immediate to Register/Memory	1 0 0 0 0 s w	mod 0 0 0 r/m	data	data if s:w = 01
Immediate to Accumulator	0 0 0 0 0 1 0 w	data	data if w = 1	
<b>ADC = Add with Carry:</b>				
Register/Memory with Register to Either	0 0 0 1 0 0 d w	mod reg r/m		
Immediate to Register/Memory	1 0 0 0 0 s w	mod 0 1 0 r/m	data	data if s:w = 01
Immediate to Accumulator	0 0 0 1 0 1 0 w	data	data if w = 1	
<b>INC = Increment:</b>				
Register/Memory	1 1 1 1 1 1 1 w	mod 0 0 0 r/m		
Register	0 1 0 0 0 reg			
<b>AAA = ASCII Adjust for Add</b>	0 0 1 1 0 1 1 1			
<b>DAA = Decimal Adjust for Add</b>	0 0 1 0 0 1 1 1			
<b>SUB = Subtract:</b>				
Register/Memory and Register to Either	0 0 1 0 1 0 d w	mod reg r/m		
Immediate from Register/Memory	1 0 0 0 0 s w	mod 1 0 1 r/m	data	data if s:w = 01
Immediate from Accumulator	0 0 1 0 1 1 0 w	data	data if w = 1	
<b>SBB = Subtract with Borrow:</b>				
Register/Memory and Register to Either	0 0 0 1 1 0 d w	mod reg r/m		
Immediate from Register/Memory	1 0 0 0 0 s w	mod 0 1 1 r/m	data	data if s:w = 01
Immediate from Accumulator	0 0 0 1 1 1 0 w	data	data if w = 1	
<b>DEC = Decrement:</b>				
Register/Memory	1 1 1 1 1 1 1 w	mod 0 0 1 r/m		
Register	0 1 0 0 1 reg			
<b>NEG = Change Sign</b>	1 1 1 1 0 1 1 w	mod 0 1 1 r/m		
<b>CMP = Compare:</b>				
Register/Memory and Register	0 0 1 1 1 0 d w	mod reg r/m		
Immediate with Register/Memory	1 0 0 0 0 s w	mod 1 1 1 r/m	data	data if s:w = 01
Immediate with Accumulator	0 0 1 1 1 1 0 w	data	data if w = 1	
<b>AAS = ASCII Adjust for Subtract</b>	0 0 1 1 1 1 1 1			
<b>DAS = Decimal Adjust for Subtract</b>	0 0 1 0 1 1 1 1			
<b>MUL = Multiply (Unsigned)</b>	1 1 1 1 0 1 1 w	mod 1 0 0 r/m		
<b>IMUL = Integer Multiply (Signed)</b>	1 1 1 1 0 1 1 w	mod 1 0 1 r/m		
<b>AAM = ASCII Adjust for Multiply</b>	1 1 0 1 0 1 0 0	0 0 0 0 1 0 1 0		
<b>DIV = Divide (Unsigned)</b>	1 1 1 1 0 1 1 w	mod 1 1 0 r/m		
<b>IDIV = Integer Divide (Signed)</b>	1 1 1 1 0 1 1 w	mod 1 1 1 r/m		
<b>AAD = ASCII Adjust for Divide</b>	1 1 0 1 0 1 0 1	0 0 0 0 1 0 1 0		
<b>CBW = Convert Byte to Word</b>	1 0 0 1 1 0 0 0			
<b>CWD = Convert Word to Double Word</b>	1 0 0 1 1 0 0 1			

**Instruction Set Summary** (Continued)

MNEMONIC AND DESCRIPTION	INSTRUCTION CODE			
	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
<b>LOGIC</b>				
<b>NOT = Invert</b>	1 1 1 1 0 1 1 w	mod 0 1 0 r/m		
<b>SHL/SAL = Shift Logical/Arithmetic Left</b>	1 1 0 1 0 0 v w	mod 1 0 0 r/m		
<b>SHR = Shift Logical Right</b>	1 1 0 1 0 0 v w	mod 1 0 1 r/m		
<b>SAR = Shift Arithmetic Right</b>	1 1 0 1 0 0 v w	mod 1 1 1 r/m		
<b>ROL = Rotate Left</b>	1 1 0 1 0 0 v w	mod 0 0 0 r/m		
<b>ROR = Rotate Right</b>	1 1 0 1 0 0 v w	mod 0 0 1 r/m		
<b>RCL = Rotate Through Carry Flag Left</b>	1 1 0 1 0 0 v w	mod 0 1 0 r/m		
<b>RCR = Rotate Through Carry Right</b>	1 1 0 1 0 0 v w	mod 0 1 1 r/m		
<b>AND = And:</b>				
Reg./Memory and Register to Either	0 0 1 0 0 0 0 d w	mod reg r/m		
Immediate to Register/Memory	1 0 0 0 0 0 0 w	mod 1 0 0 r/m	data	data if w = 1
Immediate to Accumulator	0 0 1 0 0 1 0 w	data	data if w = 1	
<b>TEST = And Function to Flags, No Result:</b>				
Register/Memory and Register	1 0 0 0 0 1 0 w	mod reg r/m		
Immediate Data and Register/Memory	1 1 1 1 0 1 1 w	mod 0 0 0 r/m	data	data if w = 1
Immediate Data and Accumulator	1 0 1 0 1 0 0 w	data	data if w = 1	
<b>OR = Or:</b>				
Register/Memory and Register to Either	0 0 0 0 1 0 d w	mod reg r/m		
Immediate to Register/Memory	1 0 0 0 0 0 0 w	mod 1 0 1 r/m	data	data if w = 1
Immediate to Accumulator	0 0 0 0 1 1 0 w	data	data if w = 1	
<b>XOR = Exclusive or:</b>				
Register/Memory and Register to Either	0 0 1 1 0 0 d w	mod reg r/m		
Immediate to Register/Memory	1 0 0 0 0 0 0 w	mod 1 1 0 r/m	data	data if w = 1
Immediate to Accumulator	0 0 1 1 0 1 0 w	data	data if w = 1	
<b>STRING MANIPULATION</b>				
<b>REP = Repeat</b>	1 1 1 1 0 0 1 z			
<b>MOVS = Move Byte/Word</b>	1 0 1 0 0 1 0 w			
<b>CMPS = Compare Byte/Word</b>	1 0 1 0 0 1 1 w			
<b>SCAS = Scan Byte/Word</b>	1 0 1 0 1 1 1 w			
<b>LODS = Load Byte/Word to AL/AX</b>	1 0 1 0 1 1 0 w			
<b>STOS = Store Byte/Word from AL/A</b>	1 0 1 0 1 0 1 w			
<b>CONTROL TRANSFER</b>				
<b>CALL = Call:</b>				
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	
		seg-low	seg-high	
Indirect Intersegment	1 1 1 1 1 1 1 1	mod 0 1 1 r/m		

**Instruction Set Summary** (Continued)

MNEMONIC AND DESCRIPTION	INSTRUCTION CODE			
	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
<b>JMP = Unconditional Jump:</b>				
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	
Direct Intersegment	1 1 1 0 1 0 1 0	offset-low	offset-high	
		seg-low	seg-high	
Indirect Intersegment	1 1 1 1 1 1 1 1	mod 1 0 1 r/m		
<b>RET = Return from CALL:</b>				
Within Segment	1 1 0 0 0 0 1 1			
Within Seg Adding Immed 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	data-high	
<b>JE/JZ = Jump on Equal/Zero</b>	0 1 1 1 0 1 0 0	disp		
<b>JL/JNGE = Jump on Less/Not Greater or Equal</b>	0 1 1 1 1 1 0 0	disp		
<b>JLE/JNG = Jump on Less or Equal/ Not Greater</b>	0 1 1 1 1 1 1 0	disp		
<b>JB/JNAE = Jump on Below/Not Above or Equal</b>	0 1 1 1 0 0 1 0	disp		
<b>JBE/JNA = Jump on Below or Equal/Not Above</b>	0 1 1 1 0 1 1 0	disp		
<b>JP/JPE = Jump on Parity/Parity Even</b>	0 1 1 1 1 0 1 0	disp		
<b>JO = Jump on Overflow</b>	0 1 1 1 0 0 0 0	disp		
<b>JS = Jump on Sign</b>	0 1 1 1 1 0 0 0	disp		
<b>JNE/JNZ = Jump on Not Equal/Not Zero</b>	0 1 1 1 0 1 0 1	disp		
<b>JNL/JGE = Jump on Not Less/Greater or Equal</b>	0 1 1 1 1 1 0 1	disp		
<b>JNLE/JG = Jump on Not Less or Equal/Greater</b>	0 1 1 1 1 1 1 1	disp		
<b>JNB/JAE = Jump on Not Below/Above or Equal</b>	0 1 1 1 0 0 1 1	disp		
<b>JNBE/JA = Jump on Not Below or Equal/Above</b>	0 1 1 1 0 1 1 1	disp		
<b>JNP/JPO = Jump on Not Par/Par Odd</b>	0 1 1 1 1 0 1 1	disp		
<b>JNO = Jump on Not Overflow</b>	0 1 1 1 0 0 0 1	disp		
<b>JNS = Jump on Not Sign</b>	0 1 1 1 1 0 0 1	disp		
<b>LOOP = Loop CX Times</b>	1 1 1 0 0 0 1 0	disp		
<b>LOOPZ/LOOPE = Loop While Zero/Equal</b>	1 1 1 0 0 0 0 1	disp		
<b>LOOPNZ/LOOPNE = Loop While Not Zero/Equal</b>	1 1 1 0 0 0 0 0	disp		
<b>JCXZ = Jump on CX Zero</b>	1 1 1 0 0 0 1 1	disp		
<b>INT = Interrupt</b>				
Type Specified	1 1 0 0 1 1 0 1	type		
Type 3	1 1 0 0 1 1 0 0			
<b>INTO = Interrupt on Overflow</b>	1 1 0 0 1 1 1 0			
<b>IRET = Interrupt Return</b>	1 1 0 0 1 1 1 1			

**Instruction Set Summary** (Continued)

MNEMONIC AND DESCRIPTION	INSTRUCTION CODE			
	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
<b>PROCESSOR CONTROL</b>				
<b>CLC</b> = Clear Carry	1 1 1 1 1 0 0 0			
<b>CMC</b> = Complement Carry	1 1 1 1 0 1 0 1			
<b>STC</b> = Set Carry	1 1 1 1 1 0 0 1			
<b>CLD</b> = Clear Direction	1 1 1 1 1 1 0 0			
<b>STD</b> = Set Direction	1 1 1 1 1 1 0 1			
<b>CLI</b> = Clear Interrupt	1 1 1 1 1 0 1 0			
<b>STI</b> = Set Interrupt	1 1 1 1 1 0 1 1			
<b>HLT</b> = Halt	1 1 1 1 0 1 0 0			
<b>WAIT</b> = Wait	1 0 0 1 1 0 1 1			
<b>ESC</b> = Escape (to External Device)	1 1 0 1 1 x x x	mod x x x r/m		
<b>LOCK</b> = Bus Lock Prefix	1 1 1 1 0 0 0 0			

**NOTES:**

AL = 8-bit accumulator  
 AX = 16-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 values  
 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 = O†, disp-low and disp-high are absent  
 if mod = 01 then DISP = disp-low sign-extended 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.  
 †† MOV CS, REG/MEMORY not allowed.

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 11 0

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	00 ES
101 BP	101 CH	00 ES
110 SI	110 DH	00 ES
111 DI	111 BH	00 ES

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

FLAGS =

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

**Die Characteristics**

**DIE DIMENSIONS:**

6370µm x 7420µm x 485µm

**INTERFACE MATERIALS:**

**Glassivation:**

Thickness: 8kÅ ±1kÅ

**Top Metallization:**

Type: Al/Si

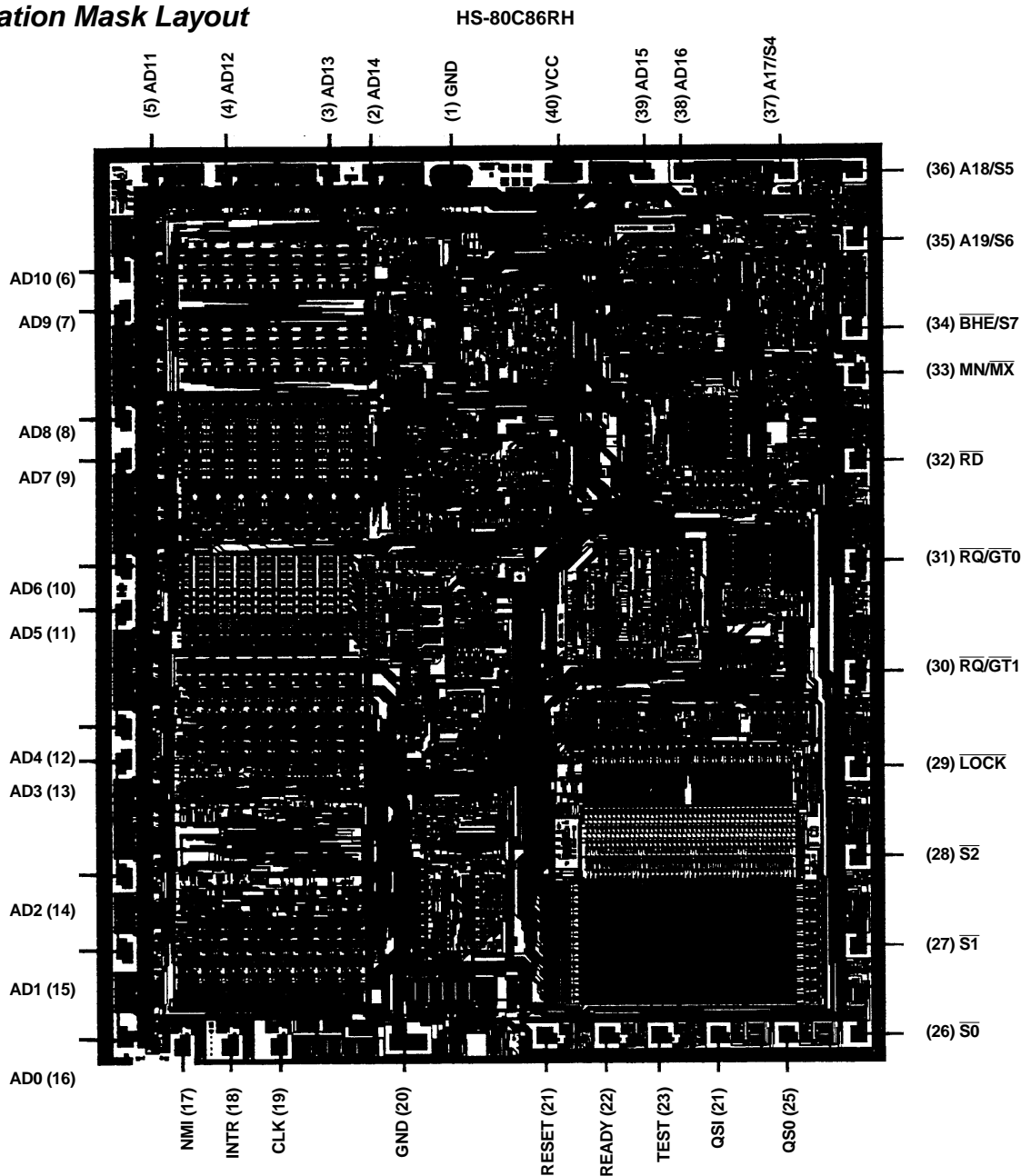
Thickness: 11kÅ ±2kÅ

**ADDITIONAL INFORMATION:**

**Worst Case Current Density:**

<2 x 10<sup>5</sup>A/cm<sup>2</sup>

**Metallization Mask Layout**



All Intersil semiconductor products are manufactured, assembled and tested under **ISO9000** quality systems certification.

*Intersil semiconductor products are sold by description only. Intersil Corporation reserves the right to make changes in circuit design and/or specifications at any time without notice. Accordingly, the reader is cautioned to verify that data sheets are current before placing orders. Information furnished by Intersil is believed to be accurate and reliable. However, no responsibility is assumed by Intersil or its subsidiaries for its use; nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Intersil or its subsidiaries.*

For information regarding Intersil Corporation and its products, see web site [www.intersil.com](http://www.intersil.com)