Order this document by:
MC68181/D, Rev. 0

MC68181

Advance Information
ROAMING FLEX™chip SIGNAL PROCESSOR

The FLEX™ protocol is a multi-speed, high-performance protocol adopted by leading service providers worldwide as a de facto paging standard. The FLEX protocol gives service providers the increased capacity, added reliability, and enhanced pager battery performance they need today. It also provides an upward migration path to the service provider that is completely transparent to the end user.

The MC68181 FLEXchip™ IC is part of a total solution available from Motorola for providing FLEX capabilities in a low-power, low-cost system. The FLEXchip simplifies implementation of a FLEX paging device by interfacing with any of several off-the-shelf paging receivers, such as the MC13150 or MC3374, and any of several off-the-shelf host microcontroller/microprocessors. The primary function of the FLEXchip is to process information received and demodulated from a FLEX-radio paging channel, select messages addressed to the paging device, and communicate the message information to the host. The host interprets the message information in an appropriate manner (numeric, alphanumeric, binary, etc.) and handles all the I/O activity. The FLEXchip IC also operates the paging receiver in an efficient power consumption mode and enables the host to operate in a low power mode when message information for the paging device is not being received. **Figure 1** shows the MC68181 functional block diagram.

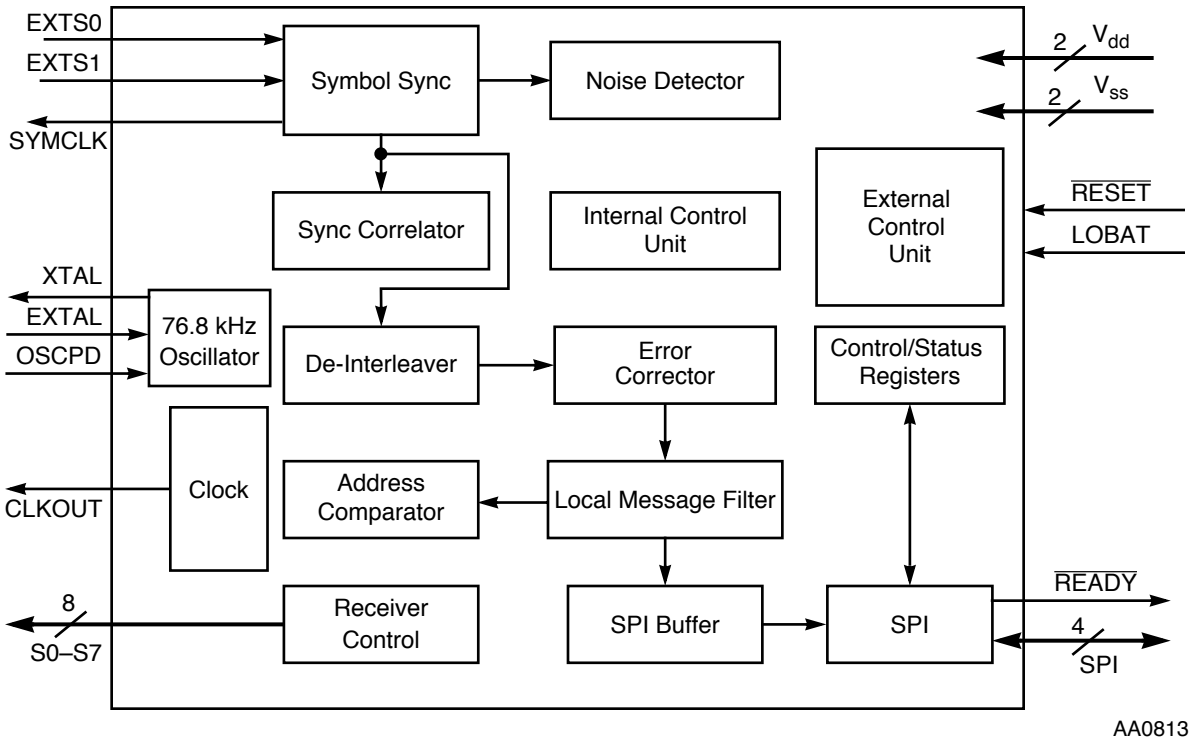


Figure 1 MC68181 Functional Block Diagram

This document contains information on a new product. Specifications and information herein are subject to change without notice.

TABLE OF CONTENTS

SECTION 1	SIGNAL/CONNECTION DESCRIPTIONS	1-1
SECTION 2	SPECIFICATIONS	2-1
SECTION 3	PACKAGING	3-1
SECTION 4	DESIGN CONSIDERATIONS	4-1
SECTION 5	ORDERING INFORMATION	5-1
APPENDIX A	FLEX SIGNAL OVERVIEW	A-1
APPENDIX B	SPI PACKET DESCRIPTION AND HANDLING	B-1
APPENDIX C	FLEXCHIP IC APPLICATION NOTES	C-1

FOR TECHNICAL ASSISTANCE:

Telephone:	1-800-521-6274
Email:	dsphelp@dsp.sps.mot.com
Internet:	http://www.motorola-dsp.com

Data Sheet Conventions

This data sheet uses the following conventions:

$\overline{\text{OVERBAR}}$	Used to indicate a signal that is active when pulled low (For example, the $\overline{\text{RESET}}$ pin is active when low.)
"asserted"	Means that a high true (active high) signal is high or that a low true (active low) signal is low
"deasserted"	Means that a high true (active high) signal is low or that a low true (active low) signal is high

Examples:

Signal/Symbol	Logic State	Signal State	Voltage
$\overline{\text{PIN}}$	True	Asserted	V_{IL}/V_{OL}
$\overline{\text{PIN}}$	False	Deasserted	V_{IH}/V_{OH}
PIN	True	Asserted	V_{IH}/V_{OH}
PIN	False	Deasserted	V_{IL}/V_{OL}

Note: Values for V_{IL}, V_{OL}, V_{IH}, and V_{OH} are defined by individual product specifications.

FEATURES

- FLEX paging protocol signal processor
- Sixteen programmable user address words
- Sixteen fixed temporary addresses
- 1600-, 3200-, and 6400-bits-per-second decoding
- Any-phase or single-phase decoding
- Uses standard Serial Peripheral Interface (SPI) in Slave mode
- Allows low current Stop mode operation of host processor
- Highly programmable receiver control
- Real-time clock time base
- FLEX software fragmentation and group messaging support
- Real time clock over-the-air update support
- Compatible with synthesized receivers
- Low Battery Indication (external detector)
- 1.8 to 3.3 V low power operation
- 32-pin Thin Quad Flat Pack (TQFP) package

ADDITIONAL SUPPORT

FLEX System Software from Motorola is a family of software components for building world-class products incorporating messaging capabilities. FLEXstack™ Software is specifically designed to support the FLEXchip IC. FLEXstack Software runs on a product's host processor and takes care of communicating with the FLEXchip IC and fully interpreting the codewords that are passed to the host from the FLEXchip IC.

DOCUMENTATION

This document is the primary document supporting the MC68181 FLEXchip IC. Documentation is available from a local Motorola distributor, a Motorola semiconductor sales office, a Motorola Literature Distribution Center, or through the Motorola DSP home page on the Internet (the source for the latest information).

SECTION 1

SIGNAL/CONNECTION DESCRIPTIONS

SIGNAL GROUPINGS

The input and output signals of the MC68181 are organized into six functional groups, as shown in **Table 1-1** and as illustrated in **Figure 1-1**.

Table 1-1 MC68181 Functional Signal Groupings

Functional Group	Number of Signals	Detailed Description
Power Input and Monitoring	7	**Table 1-2**
Processor Clocks	1	**Table 1-3**
Reset	1	**Table 1-4**
Current Symbol Inputs	2	**Table 1-5**
Serial Peripheral Interface (SPI)	5	**Table 1-6**
Receiver Control Port	8	**Table 1-7**

Figure 1-1 is a diagram of MC68181 signals by functional group.

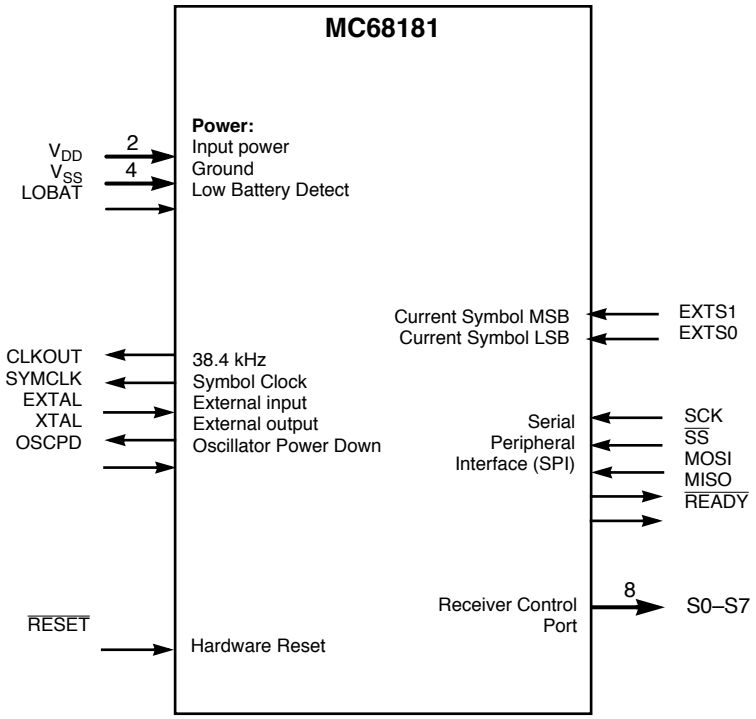


Figure 1-1 Signals Identified by Functional Group

POWER INPUT AND MONITORING

Table 1-2 Power Input, Monitoring, and Control Signals

Power Name	Description
V_{DD}	**Power**—V_{DD} is the input power for the IC.
V_{SS}	**Ground**—V_{SS} is ground connection for the IC.
LOBAT	**Low Battery**—LOBAT provides an input signal to indicate to the IC when external battery power is going low.

PROCESSOR CLOCK

Table 1-3 Processor Clock Signals

Signal Name	Type	State During Reset	Signal Description
CLKOUT	Output	Indeterminate	**Clock Output**—This is typically a 38.4 kHz clock output (derived from 76.8 kHz oscillator).
SYMCLK	Output	Indeterminate	**Recovered Symbol Clock**—Data is synchronized to the internal clock and this recovered clock output enhances lockon capability by reducing jitter from cable-induced noise.
EXTAL	Input	Input	**External Clock/Crystal Input**—EXTAL interfaces the internal crystal oscillator input to a 76.8 kHz crystal input or other external input clock.
XTAL	Output	Indeterminate	**External Clock/Crystal Output**—This is typically a 76.8 kHz clock output.
OSCPD	Input	Input	**Oscillator Power Down**—This input determines whether the internal oscillator is used. Connect this pin to V_{SS} when using the 76.8 kHz crystal input. Connect this pin to V_{DD} when using an external input clock signal.

RESET

Table 1-4 Test and Reset Signals

Signal Name	Type	State During Reset	Signal Description
$\overline{\text{RESET}}$	Input	Input	**Reset—**This input is a direct hardware reset on the FLEXchip IC. When $\overline{\text{RESET}}$ is asserted low, the FLEXchip IC is initialized and placed in the Reset state.

CURRENT SYMBOL INPUTS

Table 1-5 Interrupt and Mode Control

Signal Name	Type	State During Reset	Signal Description
EXTS1	Input	Input	**External Symbol 1**—This is the Most Significant Bit (MSB) of the symbol being tested.
EXTS0	Input	Input	**External Symbol** 0—This is the Least Significant Bit (LSB) of the symbol being tested.

SERIAL PERIPHERAL INTERFACE (SPI)

Table 1-6 Serial Peripheral Interface (SPI) Signals

Signal Name	Signal Type	State during Reset	Signal Description
SCK	Input	Input	**SPI Serial Clock**—The SCK signal is an input, and the clock signal from the external master synchronizes the data transfer. The SCK signal is ignored by the SPI if the Slave Select ($\overline{SS}$) signal is not asserted.
$\overline{SS}$	Input	Input	**SPI Slave Select**—This signal is used to enable the SPI slave for transfer.
MOSI	Input	Input	**SPI Master-Out-Slave-In**—Since the MC68181 is always a slave device, this is the data input for SPI communications. The MOSI signal is used in conjunction with the MISO signal for transmitting and receiving serial data.
MISO	Output	Tri-stated	**SPI Master-In-Slave-Out**—Since the MC68181 is always a slave device, this is the data output for SPI communications. The MISO signal is used in conjunction with the MOSI signal for transmitting and receiving serial data.
$\overline{READY}$	Output	Output, driven high	**SPI Ready**—This signal is driven low when the FLEXchip IC is ready for an SPI packet.

RECEIVER CONTROL PORT

Table 1-7 Receiver Control Port Signals

Signal Name	Signal Type	State during Reset	Signal Description
S0–S7	Output	Tri-stated	**Serial Port 0–Serial Port 7**—These signals are the eight receiver control ports.

SECTION 2

SPECIFICATIONS

INTRODUCTION

The MC68181 is fabricated in high density CMOS with Transistor-Transistor Logic (TTL) compatible inputs and outputs.

MAXIMUM RATINGS

CAUTION

This device contains circuitry protecting against damage due to high static voltage or electrical fields; however, normal precautions should be taken to avoid exceeding maximum voltage ratings. Reliability is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either GND or V_{CC}).

Note: In the calculation of timing requirements, adding a maximum value of one specification to a minimum value of another specification does not yield a reasonable sum. A maximum specification is calculated using a worst case variation of process parameter values in one direction. The minimum specification is calculated using the worst case for the same parameters in the opposite direction. Therefore, a "maximum" value for a specification will never occur in the same device that has a "minimum" value for another specification; adding a maximum to a minimum represents a condition that can never exist.

Table 2-1 Maximum Ratings

Rating[1]	Symbol	Value[1, 2]	Unit
Supply Voltage	V_{CC}	–0.3 to +3.6	V
All input voltages	V_{IN}	GND – 0.5 to V_{CC} + 0.5	V
Current drain per pin excluding V_{DD} and V_{SS}	I	10	mA
Operating temperature range	T_A	–30 to +85	°C
Storage temperature	T_{STG}	–55 to +150	°C

Note: 1. GND = 0 V, V_{CC} = 1.8 to 3.3 V, T_A = 0°C to +70°C
2. Absolute maximum ratings are stress ratings only, and functional operation at the maximum is not guaranteed. Stress beyond the maximum rating may affect device reliability or cause permanent damage to the device.

THERMAL CHARACTERISTICS

Table 2-2 Thermal Characteristics

Characteristic	Symbol	TQFP Value	Unit
Junction-to-ambient thermal resistance[1]	$R_{\theta JA}$ or θ_{JA}	95	°C/W
Thermal characterization parameter	Ψ_{JT}	21	°C/W

Note: 1. Junction-to-ambient thermal resistance is based on measurements on a horizontal, single-sided Printed Circuit Board per SEMI G38-87 in natural convection.(SEMI is Semiconductor Equipment and Materials International, 805 East Middlefield Rd., Mountain View, CA 94043, (415) 964-5111) Values were measured with the parts mounted on thermal test boards meeting the specification EIA/JESD51-3.

DC ELECTRICAL CHARACTERISTICS

Table 2-3 DC Electrical Characteristics

Characteristics	Symbol	Min	Typ	Max	Unit
Supply voltage	V_{CC}	1.8	2.0 or 3.3	3.6	V
Input high voltage $\overline{RESET}$, $\overline{SS}$, SCK, MOSI All other inputs	V_{IH}	 $0.75 \times V_{DD}$ $0.7 \times V_{DD}$	 — —	 V_{CC} V_{CC}	 V V
Input low voltage	V_{IL}	—	—	$0.2 \times V_{DD}$	V
Input leakage current	I_{IN}	–0.25	—	0.25	µA
High impedance (off-state) input current (@ 1.44 V /0.3 V)	I_{TSI}	–10	—	+10	µA
Output high voltage (I_{OH} = –1.0 mA)	V_{OH}	$0.8 \times V_{DD}$	—	—	V
Output low voltage (I_{OL} = 2.8 mA)	V_{OL}	—	—	0.3	V
Internal Supply Current[1]	I_{CC}	—	100	—	µA
Input capacitance	C_{IN}	—	10	—	pF

Note: 1. This value is for static I_{CC}.

AC ELECTRICAL CHARACTERISTICS

The timing waveforms in the AC Electrical Characteristics are tested with a V_{IL} maximum of $0.2 \times V_{DD}$ in V and a V_{IH} minimum of $0.7 \times V_{DD}$ in V for all inputs. AC timing specifications that are referenced to a device input signal are measured in production with respect to the 50% point of the respective input signal's transition. MC68181 output levels are measured with the production test machine V_{OL} and V_{OH} reference levels set at $0.3 \times V_{DD}$ in V and $0.6 \times V_{DD}$ in V, respectively.

INITIALIZATION TIMING

(VCC = 1.8 to 3.6 V, TA = –30 to + 85°C)

Table 2-4 Initialization Timing

Characteristic	Conditions	Symbol	Min	Max	Unit
Oscillator Start-up Time	—	t_{START}	—	5	sec
$\overline{\text{RESET}}$ Hold Time	—	t_{RESET}	200	—	ns
$\overline{\text{RESET}}$ High to $\overline{\text{READY}}$ Low	—	t_{RHRL}	76,800	76,800	T
Oscillator Warmed Up to $\overline{\text{READY}}$ Low	C_L = 50pf	t_{OWRL}	—	1	sec

Note: T is one period of the 76.8 kHz clock source. From power-up, the oscillator start-up time can impact the availability and period of clock strobes. This can affect the actual $\overline{\text{RESET}}$ high to $\overline{\text{READY}}$ low timing.

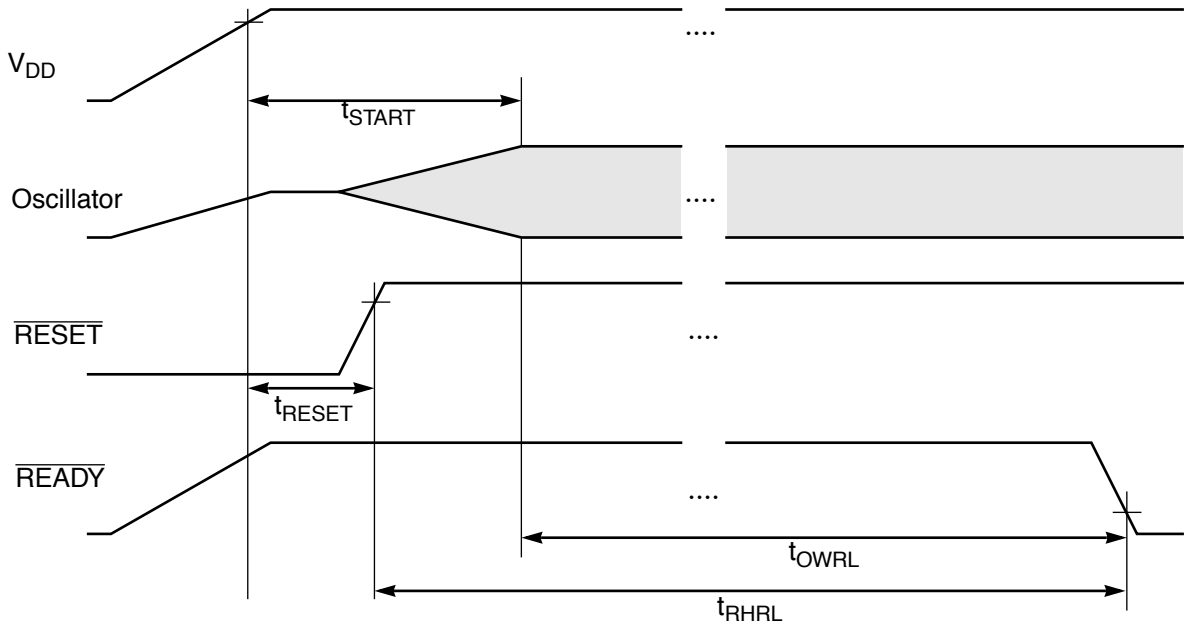


Figure 2-1 Startup Timing

RESET TIMING

(VCC = 1.8 to 3.6 V, TA = –30 to 85°C)

Table 2-5 Reset Timing

Characteristic	Conditions	Symbol	Min	Max	Unit
$\overline{\text{RESET}}$ Pulse Width	—	t_{RL}	200	—	ns
$\overline{\text{RESET}}$ Low to $\overline{\text{READY}}$ High	—	t_{RLRH}	—	200	ns
$\overline{\text{RESET}}$ High to $\overline{\text{READY}}$ Low	Requires stable 76.8 kHz clock source	t_{RHRL}	—	1	sec

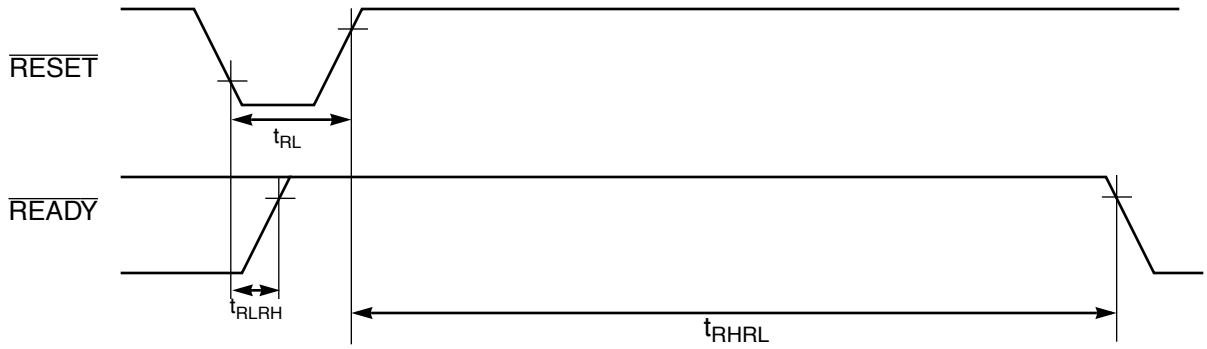


Figure 2-2 Reset Timing

SERIAL PERIPHERAL INTERFACE (SPI) TIMING

(V_{CC} = 1.8 to 3.6 V, T_A = –30 to +85°C)

Table 2-6 SPI Timing

Characteristic	Conditions	Symbol	Min	Max	Unit
Operating Frequency	—	f_{OP}	0	1	MHz
Cycle Time	—	t_{CYC}	1000	—	ns
Select Lead Time	—	t_{LEAD1}	200	—	ns
De-select Lag Time	—	t_{LAG1}	200	—	ns
Select-to-Ready Time	Previous packet did not program an address word; C_L = 50 pf	t_{RDY}	—	80	µs
Select-to-Ready Time	Previous packet programmed an address word; C_L = 50 pf	t_{RDY}	—	420	µs
Ready High Time	—	t_{RH}	50	—	µs
Ready Lead Time	—	t_{LEAD2}	200	—	ns
Not Ready Lag Time	C_L = 50pf	t_{LAG2}	—	200	ns
MOSI Data Setup Time	—	t_{SU}	200	—	ns
MOSI Data Hold Time	—	t_{HI}	200	—	ns
MISO Access Time	C_L = 50pf	t_{AC}	0	200	ns
MISO Disable Time	—	t_{DIS}	—	300	ns
MISO Data Valid Time	C_L = 50pf	t_V	—	200	ns
MISO Data Hold Time	—	t_{HO}	0	—	ns
$\overline{SS}$ High Time	—	t_{SSH}	200	—	ns
SCK High Time	—	t_{SCKH}	300	—	ns
SCK Low Time	—	t_{SCKL}	300	—	ns
SCK Rise Time	20% to 70% V_{DD}	t_R		1	µs
SCK Fall Time	20% to 70% V_{DD}	t_F		1	µs

Note: When the host reprograms an address word with a Host-to-FLEXchip packet ID > 127 (decimal), there may be an added delay before FLEXchip is ready for another packet.

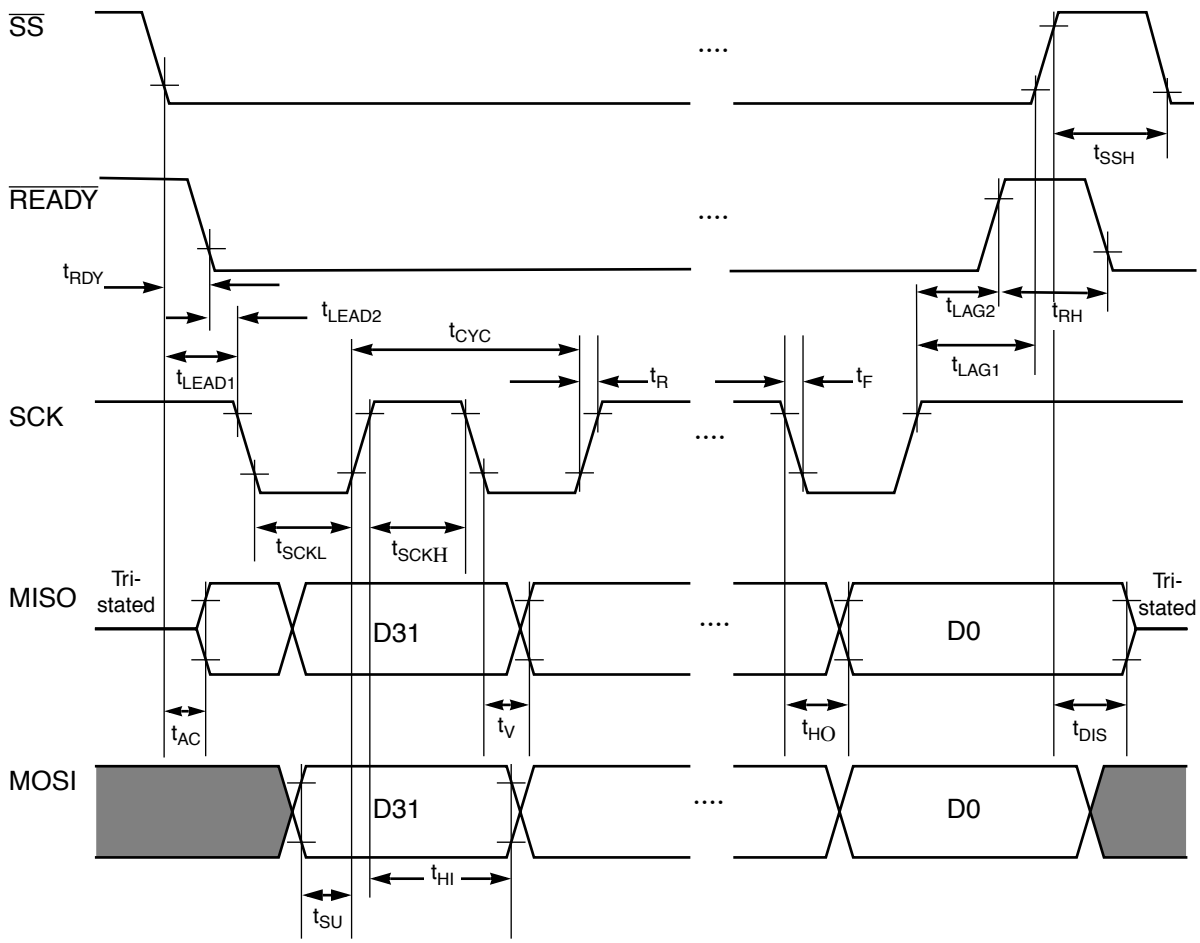


Figure 2-3 SPI Timing

SECTION 3

PACKAGING

PIN-OUT AND PACKAGE INFORMATION

This sections provides information about the available packages for this product, including diagrams of the package pinouts and tables describing how the signals described in **Section 1** are allocated. The MC68181 is available in a 32-pin Thin Quad Flat Pack (TQFP) package.

TQFP Package Description

Top and bottom views of the TQFP package are shown in **Figure 3-1** and **Figure 3-2** with their pin-outs.

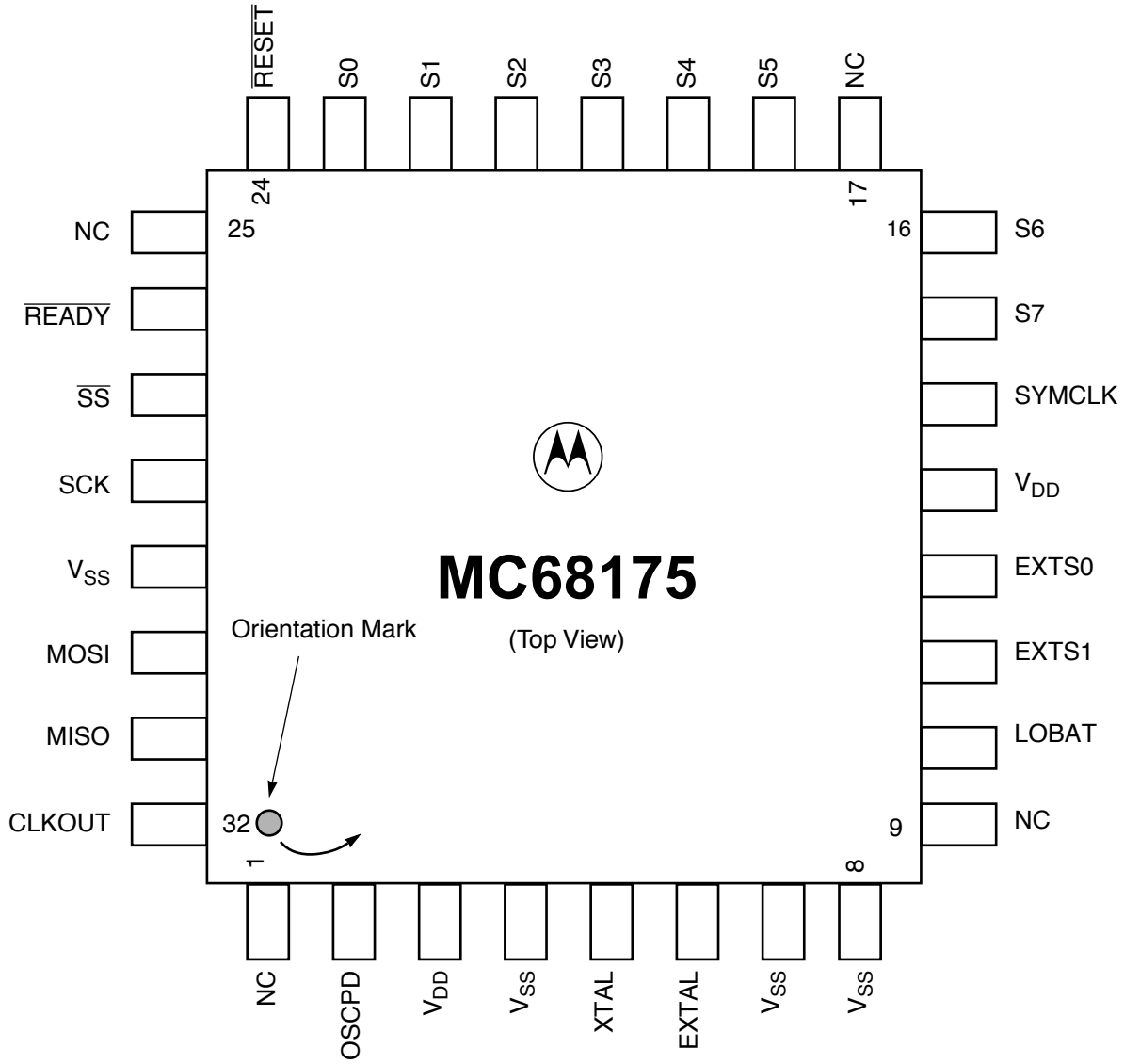


Figure 3-1 MC68181 Thin Quad Flat Pack (TQFP), Top View

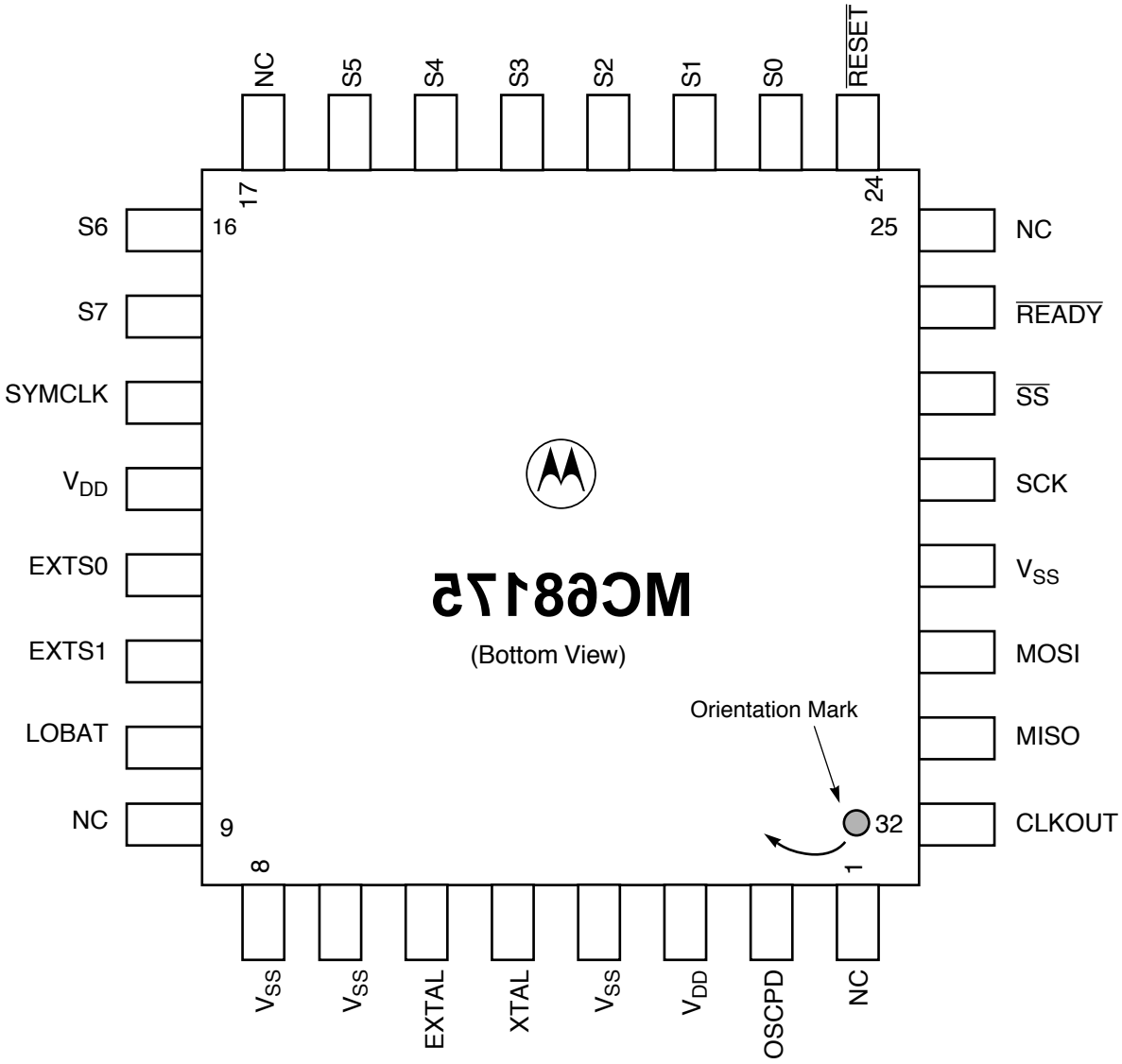


Figure 3-2 MC68181 Thin Quad Flat Pack (TQFP), Bottom View

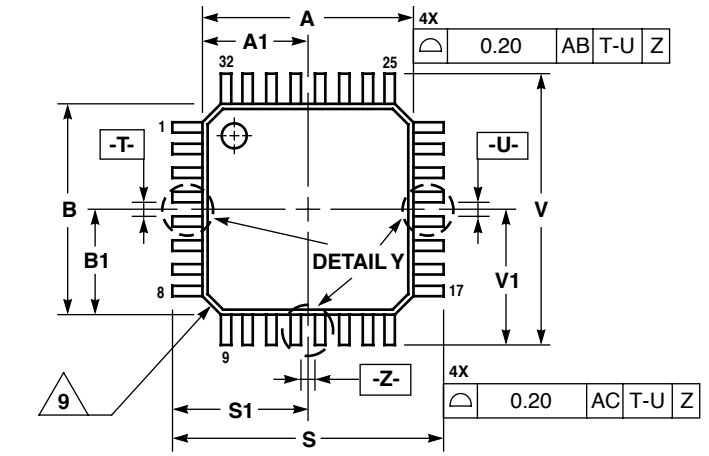
Table 3-1 Signal by Pin Number

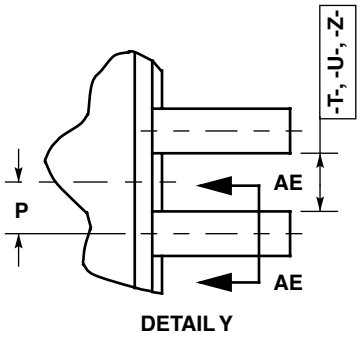
Pin #	Signal Name	Pin #	Signal Name	Pin #	Signal Name	Pin #	Signal Name
1	NC[1]	9	NC[1]	17	NC[1]	25	NC[1]
2	OSCPD	10	LOBAT	18	S5	26	$\overline{\text{READY}}$
3	V_{DD}	11	EXTS1	19	S4	27	$\overline{\text{SS}}$
4	V_{SS}[2]	12	EXTS0	20	S3	28	SCK
5	XTAL	13	V_{DD}	21	S2	29	V_{SS}[2]
6	EXTAL	14	SYMCLK	22	S1	30	MOSI
7	V_{SS}[2]	15	S7	23	S0	31	MISO
8	V_{SS}[2]	16	S6	24	$\overline{\text{RESET}}$	32	CLKOUT

Note: 1. NC indicates reserved pins. These pins must not be connected to any external line.
2. To ensure proper chip operation, all V_{SS} pins must be connected to GND.

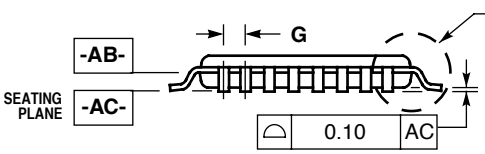
Table 3-2 Signal by Name

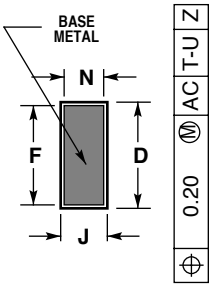
Signal Name	Pin #	Signal Name	Pin #	Signal Name	Pin #	Signal Name	Pin #
CLKOUT	32	NC	9	S2	21	SYMCLK	14
EXTAL	6	NC	17	S3	20	V_{DD}	3
EXTS0	12	NC	25	S4	19	V_{DD}	13
EXTS1	11	OSCPD	2	S5	18	V_{SS}	4
LOBAT	10	$\overline{\text{READY}}$	26	S6	16	V_{SS}	7
MISO	31	$\overline{\text{RESET}}$	24	S7	15	V_{SS}	8
MOSI	30	S0	23	SCK	28	V_{SS}	29
NC	1	S1	22	$\overline{\text{SS}}$	27	XTAL	5



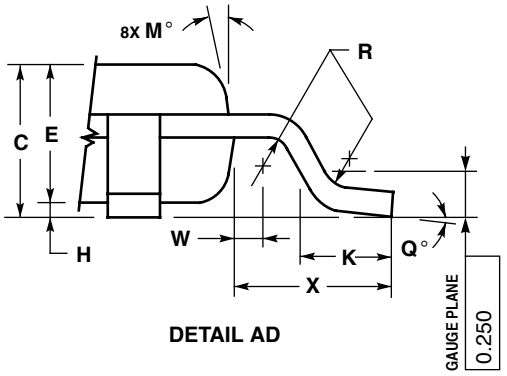


DETAIL Y





SECTION AE-AE



DETAIL AD

NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DATUM PLANE -AB- IS LOCATED AT BOTTOM OF LEAD AND IS COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.
4. DATUMS -T-, -U-, AND -Z- TO BE DETERMINED AT DATUM PLANE -AB-.
5. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE -AC-.
6. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.250 (0.010) PER SIDE. DIMENSIONS A AND B DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE -AB-.
7. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. DAMBAR PROTRUSION SHALL NOT CAUSE THE D DIMENSION TO EXCEED 0.520.

	MILLIMETERS	
DIM	MIN	MAX
A	7.000 BSC	
A1	3.500 BSC	
B	7.000 BSC	
B1	3.500 BSC	
C	1.400	1.600
D	0.300	0.450
E	1.350	1.450
F	0.300	0.400
G	0.800 BSC	
H	0.050	0.150
J	0.090	0.200
K	0.500	0.700
M	12°□ REF	
N	0.090	0.160
P	0.400 BSC	
Q	1°□□	5°□
R	0.150	0.250
S	9.000 BSC	
S1	4.500 BSC	
V	9.000 BSC	
V1	4.500 BSC	
W	0.200 REF	
X	1.000 REF	

CASE 873A-02
ISSUE A

DATE 12/16/93

Figure 3-3 32-pin Thin Quad Flat Pack (TQFP) Mechanical Information

ORDERING DRAWINGS

Complete mechanical information regarding MC68181 packaging is available by facsimile through Motorola's Mfax™ system. Call the following number to obtain information by facsimile:

(602) 244-6591

The Mfax automated system requests the following information:

- The receiving facsimile telephone number including area code or country code
- The caller's Personal Identification Number (PIN)

Note: For first time callers, the system provides instructions for setting up a PIN, which requires entry of a name and telephone number.

- The type of information requested:
  - Instructions for using the system
  - A literature order form
  - Specific part technical information or data sheets
  - Other information described by the system messages

A total of three documents may be ordered per call.

The MC68181 32-pin TQFP package mechanical drawing is referenced as 873A-02.

SECTION 4

DESIGN CONSIDERATIONS

THERMAL DESIGN CONSIDERATIONS

An estimation of the chip junction temperature, T_J, in °C can be obtained from the equation:

Equation 1: $T_J = T_A + (P_D \times R_{\theta JA})$

Where:

T_A = ambient temperature ˚C
$R_{\theta JA}$ = package junction-to-ambient thermal resistance ˚C/W
P_D = power dissipation in package

Historically, thermal resistance has been expressed as the sum of a junction-to-case thermal resistance and a case-to-ambient thermal resistance:

Equation 2: $R_{\theta JA} = R_{\theta JC} + R_{\theta CA}$

Where:

$R_{\theta JA}$ = package junction-to-ambient thermal resistance ˚C/W
$R_{\theta JC}$ = package junction-to-case thermal resistance ˚C/W
$R_{\theta CA}$ = package case-to-ambient thermal resistance ˚C/W

$R_{\theta JC}$ is device-related and cannot be influenced by the user. The user controls the thermal environment to change the case-to-ambient thermal resistance, $R_{\theta CA}$. For example, the user can change the air flow around the device, add a heat sink, change the mounting arrangement on the Printed Circuit Board, or otherwise change the thermal dissipation capability of the area surrounding the device on a Printed Circuit Board. This model is most useful for ceramic packages with heat sinks; some 90% of the heat flow is dissipated through the case to the heat sink and out to the ambient environment. For ceramic packages, in situations where the heat flow is split between a path to the case and an alternate path through the Printed Circuit Board, analysis of the device thermal performance may need the additional modeling capability of a system level thermal simulation tool.

The thermal performance of plastic packages is more dependent on the temperature of the Printed Circuit Board to which the package is mounted. Again, if the estimations obtained from $R_{\theta JA}$ do not satisfactorily answer whether the thermal performance is adequate, a system level model may be appropriate.

A complicating factor is the existence of three common ways for determining the junction-to-case thermal resistance in plastic packages:

- To minimize temperature variation across the surface, the thermal resistance is measured from the junction to the outside surface of the package (case) closest to the chip mounting area when that surface has a proper heat sink.
- To define a value approximately equal to a junction-to-board thermal resistance, the thermal resistance is measured from the junction to where the leads are attached to the case.
- If the temperature of the package case (T_T) is determined by a thermocouple, the thermal resistance is computed using the value obtained by the equation $(T_J - T_T)/P_D$.

As noted above, the junction-to-case thermal resistances quoted in this data sheet are determined using the first definition. From a practical standpoint, that value is also suitable for determining the junction temperature from a case thermocouple reading in forced convection environments. In natural convection, using the junction-to-case thermal resistance to estimate junction temperature from a thermocouple reading on the case of the package will estimate a junction temperature slightly hotter than actual temperature. Hence, the new thermal metric, Thermal Characterization Parameter or Ψ_{JT}, has been defined to be $(T_J - T_T)/P_D$. This value gives a better estimate of the junction temperature in natural convection when using the surface temperature of the package. Remember that surface temperature readings of packages are subject to significant errors caused by inadequate attachment of the sensor to the surface and to errors caused by heat loss to the sensor. The recommended technique is to attach a 40-gauge thermocouple wire and bead to the top center of the package with thermally conductive epoxy.

APPLICATION DESIGN CONSIDERATIONS

The FLEXchip IC connects to a receiver capable of converting a four-level audio signal into a 2-bit digital signal. The FLEXchip IC has eight receiver control lines used for warming up and shutting down a receiver in stages. The FLEXchip IC has dual bandwidth control signals for two post detection filter bandwidths for receiving the two symbol rates of the FLEX signal. The FLEXchip IC has the ability to detect a low battery signal during the receiver control sequences. It interfaces to a host MCU through a standard SPI. It has a 38.4 kHz clock output capable of driving other devices. It has a 1 minute timer that offers low power support for time of day function on the host. **Figure 4-1** shows a typical application block diagram.

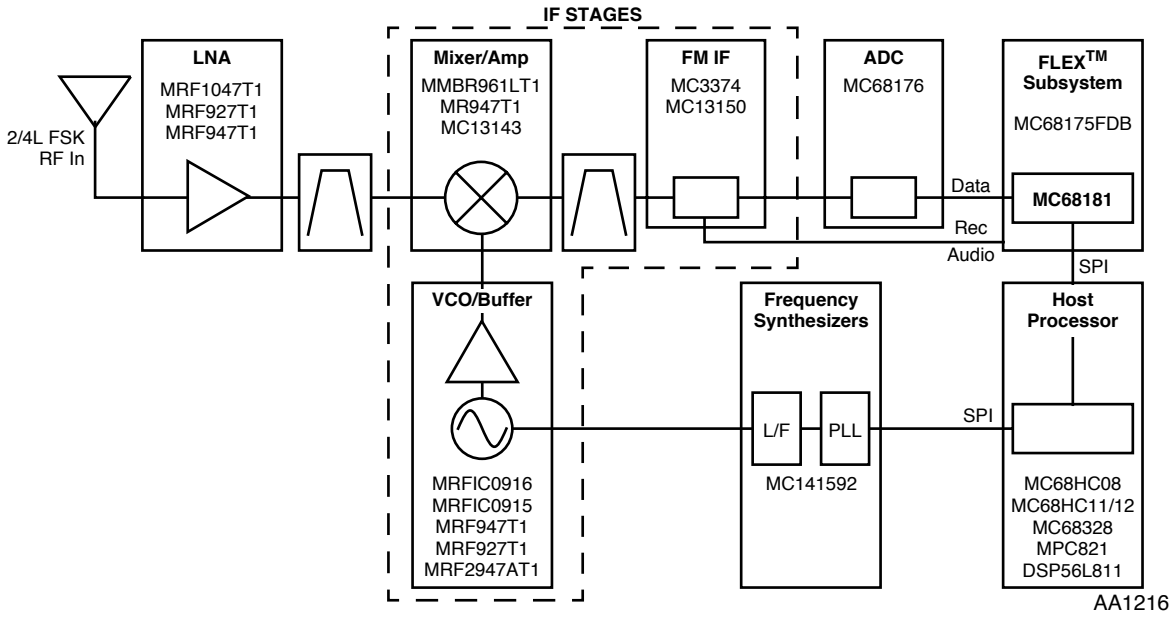
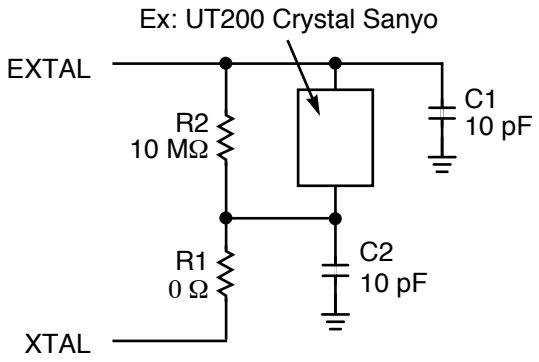


Figure 4-1 Roaming FLEX™ chip System Block Diagram

Figure 4-2 shows a recommended circuit for a 76.8 kHz crystal input.



Note: R1 can be increased in size to be used as a current limiter, if needed.

AA1069

Figure 4-2 Input Circuit for 76.8 kHz Crystal

Appendix A of this document provides a background of the FLEX signal protocol. **Appendix B** provides a description of the way in which the MC68181 FLEXchip IC handles packets through the SPI, including sections that describe transfer from the host to the decoder from the decoder to the host. **Appendix C** provides a sample application to illustrate how the MC68181 FLEXchip IC might be used in an application.

SECTION 5

ORDERING INFORMATION

Consult a Motorola Semiconductor sales office or authorized distributor to determine product availability and to place an order.

Table 5-1 Ordering Information

Part	Supply Voltage	Package Type	Pin Count	Frequency (MHz)	Order Number
MC68181	2/3 V	Thin Quad Flat Pack (TQFP)	32	1	MC68181FA

APPENDIX A
FLEX OVERVIEW

FLEX SIGNAL STRUCTURE

As shown in **Figure A-1**, a FLEX signal is transmitted on a radio channel and consists of a series of four-minute cycles, each cycle having 128 frames at 1.875 seconds per frame. A pager may be assigned to process any number of these frames. Any unassigned frames are not processed, thus reducing power required for signal processing and extending battery life. If required, however, the pager may temporarily process more complex information, because individual FLEX cycles can assign additional frames dynamically using collapse, fragmentation, temporary addressing, or carry-on information within the FLEX signal.

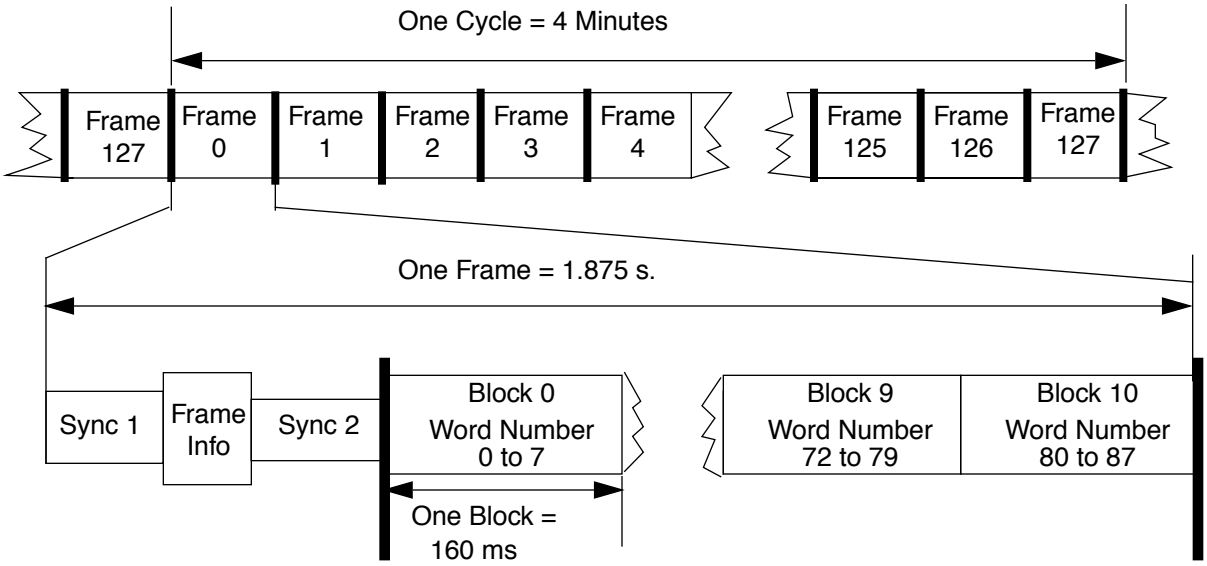


Figure A-1 FLEX™ Signal Structure

FLEX FRAME STRUCTURE

As shown in **Figure A-1** on page -1, each FLEX frame consists of:

- Synchronization portion
- Data portion—Eleven data blocks lasting 160 milliseconds each

Frame Synchronization Portion

The synchronization portion consists of:

- First synchronization signal at 1600 bps
- Frame Information Word including:
  - Frame Number 0–127 (7 bits)
  - Cycle Number 0–14 (4 bits)
- Second synchronization signal at the data rate of the interleaved portion.

FIRST SYNCHRONIZATION SIGNAL

The first synchronization signal is transmitted at 1600 bps and provides a signal to lock onto the specific frame.

FRAME INFORMATION WORD

The Frame Information Word transmits 11 bits that are divided into a 7-bit frame number and a 4-bit cycle number. This allows the pager to identify the frame and the cycle in which it resides uniquely.

SECOND SYNCHRONIZATION SIGNAL

The second synchronization signal indicates the rate at which the data portion is transmitted, 1600, 3200 or 6400 bits per second.

The 1600 bps rate is transmitted as a single phase of information (A), as shown in **Figure A-2**, at 1600 symbols per second using 2-level Frequency Shift Keyed (FSK) modulation.

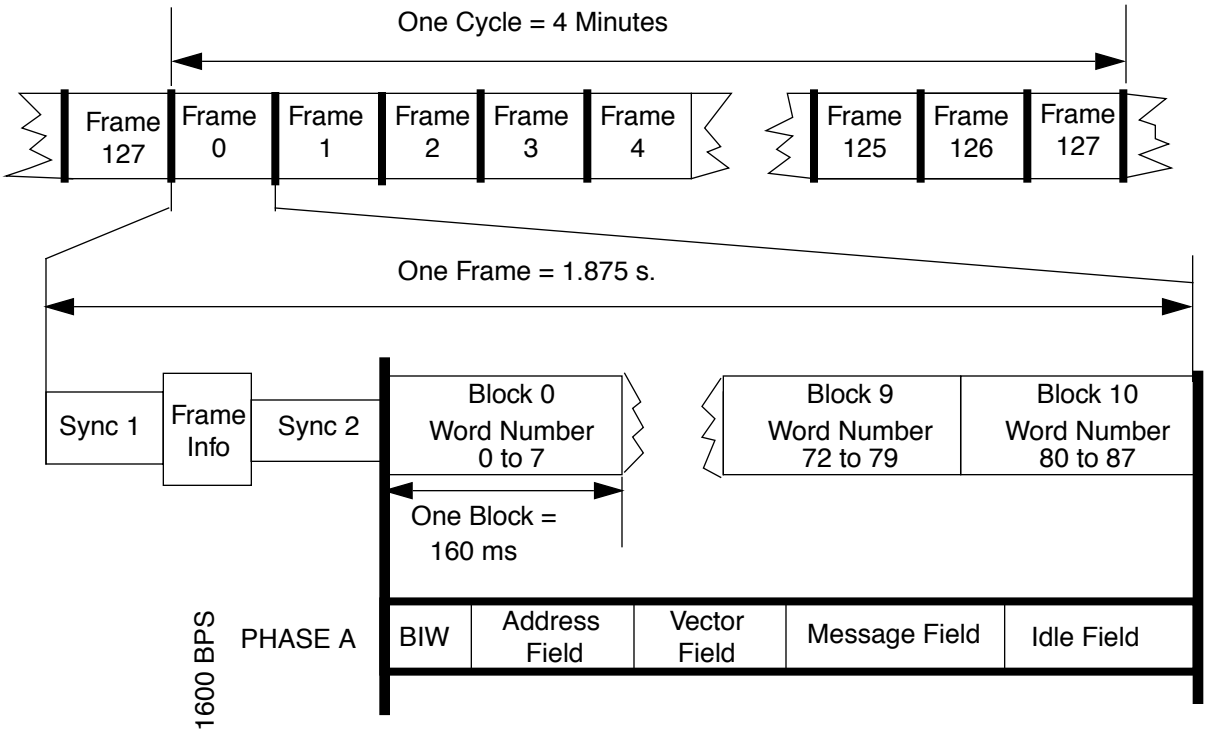


Figure A-2 FLEX™ Signal Structure for 1600 BPS

The 3200 bps rate is transmitted as two concurrent phases of information (A and C), as shown in **Figure A-3**, at either:

- 1600 symbols per second using 4-level FSK modulation, or
- 3200 symbols per second using 2-level FSK modulation.

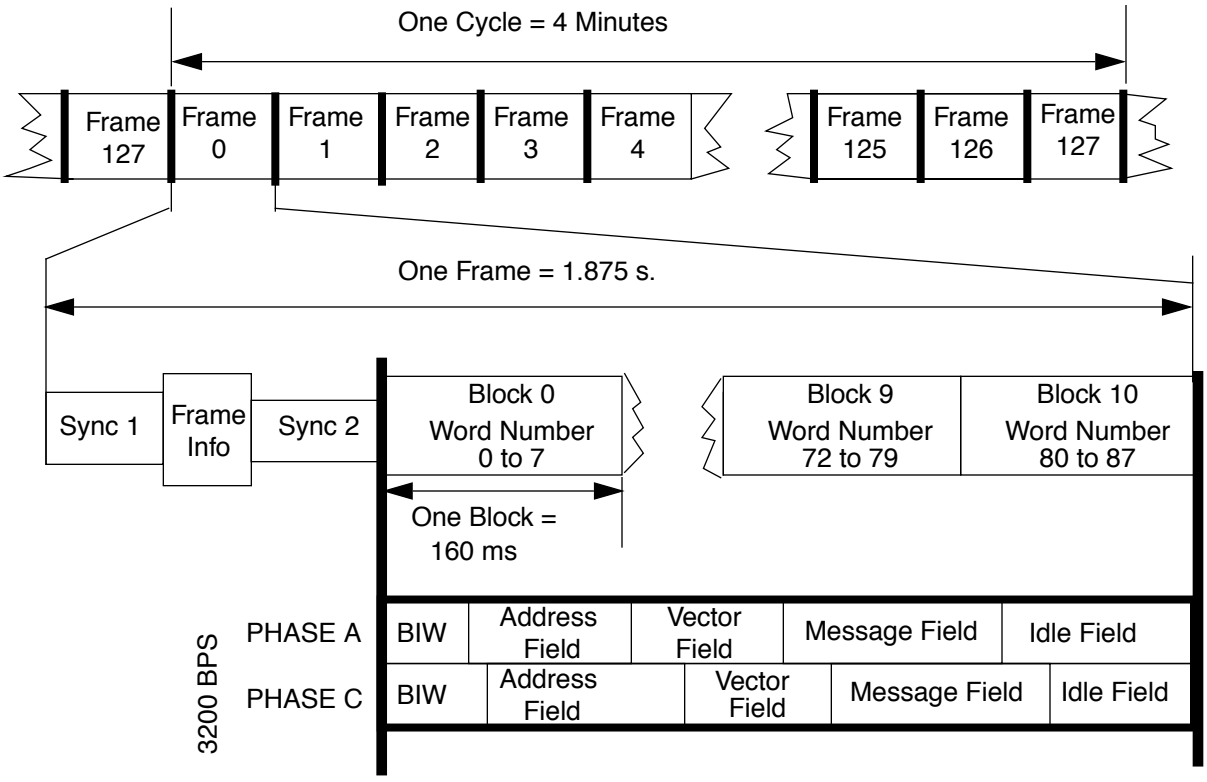


Figure A-3 FLEX™ Signal Structure for 3200 BPS

The 6400 bps rate is transmitted as four concurrent phases of information (A,B, C, and D), as shown in **Figure A-4**, at 3200 symbols per second using 4-level FSK modulation.

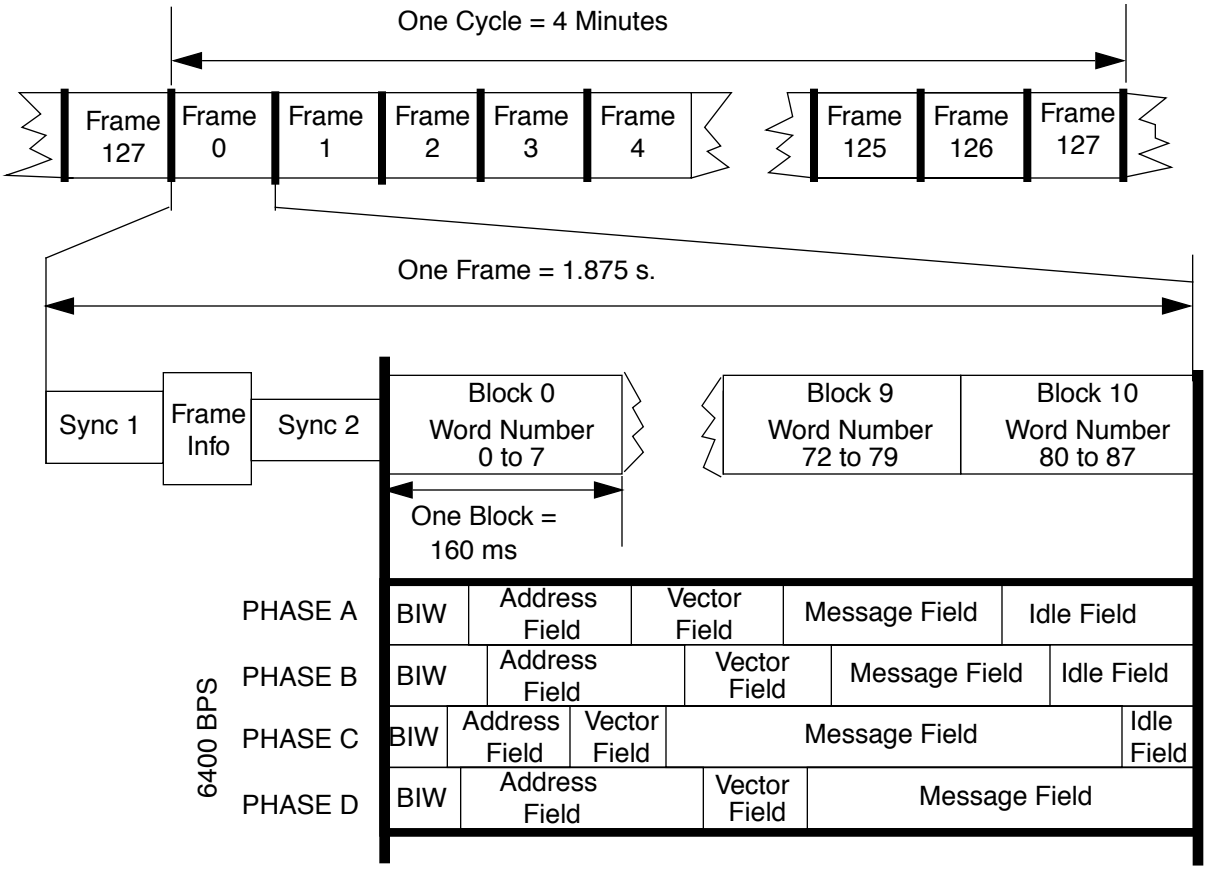


Figure A-4 FLEX™ Signal Structure For 6400 BPS

Frame Data Portion

As noted above, there are eleven data blocks following the frame synchronization portion of each frame. Each block has eight interleaved words per phase, numbered 0–87 contiguously for all eleven blocks, in every frame. Each word has information that allows for bit error correction and detection contained within an error correcting code.

Each of the eighty-eight words in each phase is organized into the following five fields:

- Block information field
- Address field
- Vector field
- Message field
- Idle field

The boundaries between the fields are independent of the block boundaries. Furthermore, at 3200 and 6400 bps, the information in one phase is independent of the information in a concurrent phase, and the boundaries between the fields of one phase are unrelated to the boundaries between the fields in a concurrent phase.

BLOCK INFORMATION FIELD

The block information field may contain information words for determining time and date information and certain paging system information.

ADDRESS FIELD

The address field contains addresses assigned to paging devices. Addresses are used to identify information sent to individual paging devices and/or groups of paging devices. An address may be either a “short” one word address or a “long” two word address. Information in the FLEX signal may indicate that an address is a priority address. An address may be a “tone only” address, in which case there is no additional information associated with the address.

VECTOR FIELD

The vector field consists of a series of vector words. Depending upon the type of message, a vector word (or words in the case of a long address) may either contain all of the information necessary for the message, or indicate the location of message words in the message field comprising the message information. If an address is not a tone only address, then there is an associated vector word in the vector field. Information in the FLEX signal indicates the location of the vector word. Short addresses have one associated vector word and long addresses two associated vector words. A pager may go to low power mode at the end of the address field if its address(es) is (are) not detected, thus resulting in battery savings.

MESSAGE FIELD

The message field consists of a series of information words containing message information. The message information may be formatted in ASCII, BCD, or binary depending upon the message type. The following sections provide a detailed description of the various types of information words that may be used in the message field.

IDLE FIELD

The idle field is used to separate blocks.

FLEX MESSAGE WORD DEFINITIONS

Numeric Data Message

The following tables describe the bit format of the numeric messages. The 4-bit numeric characters of the message are designated as lower case letters a, b, c, d, etc.

Table A-1 Standard (V = 011) or Special Format (V = 100) 4, 10, 15, 20, 25, 31, 36, or 41 Characters

Message Word	i0	i1	i2	i3	i4	i5	i6	i7	i8	i9	i10	i11	i12	i13	i14	i15	i16	i17	i18	i19	i20
1st	K_4	K_5	a_0	a_1	a_2	a_3	b_0	b_1	b_2	b_3	c_0	c_1	c_2	c_3	d_0	d_1	d_2	d_3	e_0	e_1	e2
2nd	e_3	f_0	f_1	f_2	f_3	g_0	g_1	g_2	g_3	h_0	h_1	h_2	h_3	i_0	i_1	i_2	i_3	j_0	j_1	j_2	j_3
3rd	k_0	k_1	k_2	k_3	l_0	l_1	l_2	l_3	m_0	m_1	m_2	m_3	n_0	n_1	n_2	n_3	o_0	o_1	o_2	o_3	q_0
4th	q_1	q_2	q_3	r_0	r_1	r_2	r_3	s_0	s_1	s_2	s_3	t_0	t_1	t_2	t_3	u_0	u_1	u_2	u_3	v_0	v_1
5th	v_2	v_3	w_0	w_1	w_2	w_3	y_0	y_1	y_2	y_3	z_0	z_1	z_2	z_3	A_0	A_1	A_2	A_3	B_0	B_1	B_2
6th	B_3	C_0	C_1	C_2	C_3	D_0	D_1	D_2	D_3	E_0	E_1	E_2	E_3	F_0	F_1	F_2	F_3	G_0	G_1	G_2	G_3
7th	H_0	H_1	H_2	H_3	I_0	I_1	I_2	I_3	J_0	J_1	J_2	J_3	V_0	V_1	V_2	V_3	L_0	L_1	L_2	L_3	M_0
8th	M_1	M_2	M_3	O_0	O_1	O_2	O_3	P_0	P_1	P_2	P_3	Q_0	Q_1	Q_2	Q_3	T_0	T_1	T_2	T_3	U_0	U_1

Table A-2 Numbered (V = 111) 2, 8, 13, 18, 23, 29, 34, or 39 Numeric Characters

Message Word	i0	i1	i2	i3	i4	i5	i6	i7	i8	i9	i10	i11	i12	i13	i14	i15	i16	i17	i18	i19	i20
1st	K_4	K_5	N_0	N_1	N_2	N_3	N_4	N_5	R_0	S_0	a_0	a_1	a_2	a_3	b_0	b_1	b_2	b_3	c_0	c_1	c_2
2nd	c_3	d_0	d_1	d_2	d_3	e_0	e_1	e_2	e_3	f_0	f_1	f_2	f_3	g_0	g_1	g_2	g_3	h_0	h_1	h_2	h_3
3rd	i_0	i_1	i_2	i_3	j_0	j_1	j_2	j_3	k_0	k_1	k_2	k_3	l_0	l_1	l_2	l_3	m_0	m_1	m_2	m_3	n_0
4th	n_1	n_2	n_3	o_0	o_1	o_2	o_3	q_0	q_1	q_2	q_3	r_0	r_1	r_2	r_3	s_0	s_1	s_2	s_3	t_0	t_1
5th	t_2	t_3	u_0	u_1	u_2	u_3	v_0	v_1	v_2	v_3	w_0	w_1	w_2	w_3	y_0	y_1	y_2	y_3	z_0	z_1	z_2
6th	z_3	A_0	A_1	A_2	A_3	B_0	B_1	B_2	B_3	C_0	C_1	C_2	C_3	D_0	D_1	D_2	D_3	E_0	E_1	E_2	E_3
7th	F_0	F_1	F_2	F_3	G_0	G_1	G_2	G_3	H_0	H_1	H_2	H_3	I_0	I_1	I_2	I_3	J_0	J_1	J_2	J_3	V
8th	V_1	V_2	V_3	L_0	L_1	L_2	L_3	M_0	M_1	M_2	M_3	O_0	O_1	O_2	O_3	P_0	P_1	P_2	P_3	Q_0	Q_1

Table A-3 Numeric Message Bit Definitions

Symbol	Definition
K	**6-bit Message Check Character (First 4 bits are in the vector word)**—This check character is calculated by initializing the message check character (**K**) to 0 and summing the information bits of each code word in the message, (including control information and termination characters and bits in the last message word) to a check sum register. The information bits of each word are broken into three groups: the first is the 8 bits comprising $\mathbf{i_0}$ through $\mathbf{i_7}$, the second group comprises bits $\mathbf{i_8}$ through $\mathbf{i_{15}}$, and the third group comprises bits $\mathbf{i_{16}}$ through $\mathbf{i_{20}}$. Bits $\mathbf{i_0}$, $\mathbf{i_8}$, and $\mathbf{i_{16}}$ are the LSBs of each group. The binary sum is calculated, and the result is shortened to the eight Least Significant Bits. The two Most Significant Bits are shifted 6 bits to the right and summed with the six Least Significant Bits to form a new sum. This resultant sum is one's complemented with the six LSBs of the result being transmitted as the message check character.
N	**Message Number**—When the system supports message retrieval, the system controller assigns message numbers (for each paging address separately) starting at zero and progressing up to a maximum of sixty-three in consecutive order. The actual maximum roll over number is defined in the pager code plug to accommodate values set in the system infrastructure. When message numbers are not received in order, the subscriber should assume a message has been missed. The subscriber or the pager may determine the missing message number(s) allowing a request to be made for retrieval. When a normal unnumbered numeric message is received (Message Retrieval Flag = 0), it is not to be included in the missed message calculation.

Table A-3 Numeric Message Bit Definitions (Continued)

Symbol	Definition
R	**Message Retrieval Flag**—When this bit is set to 1, the pager expects to see messages numbered in order (each address numbered separately). Detection of a missing number indicates a missed message. A message received with R = 0 is allowed to be out of order and shall not cause the pager to indicate that a message has been missed.
S	**Special Format**—In the numbered message format, this bit set to 1 indicates that a special display format should be used.

MESSAGE FILL RULES

For numeric messages of thirty-six characters or less (thirty-four characters if numbered), fewer than eight code words on the channel are required. Only code words containing the numeric message are to be transmitted. The space character (hexadecimal C) should be used to fill any unused 4-bit characters in the last word and zeros to fill any remaining partial characters. The check sum is correspondingly shortened to include only the code words comprising the shortened message along with the space and fill characters used to fill in the last word.

SPECIAL FORMAT NUMERIC

Spaces and dashes as specified by the host are inserted into the received message. This feature in certain markets saves the transmission of an additional word on the channel. As an example, in the U.S. market a 10-character string (area code plus telephone number) fits into two message words; if the dashes or parentheses are to be included in the message, a third message word on the channel is required. The actual placement can be programmed into the paging device and can vary between markets.

Hex/Binary Message

The following tables describe the bit format of the Hex/Binary messages. The data of the message is designated as lower case letters a, b, c, d, etc. Hex/binary messages can be sent as fragments. The service provider has the option of dividing the message into several pieces and sending the separate pieces at any time within a given time period.

Table A-4 Vector Type V = 110 First Only Fragment

Message Word	i_0	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}	i_{11}	i_{12}	i_{13}	i_{14}	i_{15}	i_{16}	i_{17}	i_{18}	i_{19}	i_{20}
1st	K_0	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9	K_{10}	K_{11}	C_0	F_0	F_1	N_0	N_1	N_2	N_3	N_4	N_5
2nd	R_0	M_0	D_0	H_0	B_0	B_1	B_2	B_3	s_0	s_1	s_2	s_3	s_4	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7
3rd	a_0	a_1	a_2	a_3	b_0	b_1	b_2	b_3	c_0	c_1	c_2	c_3	d_0	d_1	d_2	d_3	e_0	e_1	e_2	e_3	f_0
4th	f_1	f_2	f_3	g_0	g_1	g_2	g_3	h_0	h_1	h_2	h_3	i_0	i_1	i_2	i_3	j_0	j_1	j_2	j_3	k_0	k_1
5th	k_2	k_3	l_0	l_1	l_2	l_3	m_0	m_1	m_2	m_3	n_0	n_1	n_2	n_3	o_0	o_1	o_2	o_3	q_0	q_1	q_2
6th	q_3	r_0	r_1	r_2	r_3	s_0	s_1	s_2	s_3	t_0	t_1	t_2	t_3	u_0	u_1	u_2	u_3	v_0	v_1	v_2	v_3
...																					
nth	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i

Table A-5 Vector Type V=110 All Other Fragments

Message Word	i0	i1	i2	i3	i4	i5	i6	i7	i8	i9	i10	i11	i12	i13	i14	i15	i16	i17	i18	i19	i20
1st	K_0	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9	K_{10}	K_{11}	C_0	F_0	F_1	N_0	N_1	N_2	N_3	N_4	N_5
2nd	a_0	a_1	a_2	a_3	b_0	b_1	b_2	b_3	c_0	c_1	c_2	c_3	d_0	d_1	d_2	d_3	e_0	e_1	e_2	e_3	f_0
3rd	f_1	f_2	f_3	g_0	g_1	g_2	g_3	h_0	h_1	h_2	h_3	i_0	i_1	i_2	i_3	j_0	j_1	j_2	j_3	k_0	k_1
4th	k_2	k_3	l_0	l_1	l_2	l_3	m_0	m_1	m_2	m_3	n_0	n_1	n_2	n_3	o_0	o_1	o_2	o_3	q_0	q_1	q_2
5th	q_3	r_0	r_1	r_2	r_3	s_0	s_1	s_2	s_3	t_0	t_1	t_2	t_3	u_0	u_1	u_2	u_3	v_0	v_1	v_2	v_3
...																					
nth	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i

Table A-6 Hex/Binary Message Bit Definitions

Symbol[1]	Definition
K	**12-bit Fragment Check Sum**—This check sum is calculated by initializing the Fragment Check Sum field (**K**) to 0 and calculating a sum over the information bits of each code word in the message fragment (including control information and termination characters/bits in the last fragment word). This sum requires that the information bits of each word be broken into three groups: the first is the 8 bits comprising $\mathbf{i_0}$ through $\mathbf{i_7}$, the second group comprises bits $\mathbf{i_8}$ through $\mathbf{i_{15}}$, and the third group comprises bits $\mathbf{i_{16}}$ through $\mathbf{i_{20}}$. Bits $\mathbf{i_0}$, $\mathbf{i_8}$, and $\mathbf{i_{16}}$ are the LSBs of each group. The binary sum is calculated over all code words in the fragment, the one's complement of the sum is determined, and the twelve LSBs of the result is placed into the Fragment Check Sum field to be transmitted at the beginning of the fragment.
C	**1-bit Message Continued Flag**—When set to 1, this flag indicates fragments of this message are to be expected in any or possibly all of the following frames until a fragment with C = 0 is found. The longest message that fits into a frame is eighty-four code words. Three alpha characters per word yields a maximum message of 252 characters in a frame, assuming no other traffic. Messages longer than this value must be sent as several fragments.
F	**2-bit Message Fragment Number**—This is a modulo 3 message fragment number that is incremented by 1 in successive message fragments. The initial fragment starts at 11 and each following fragment is incremented by 1 modulo 3, (11, 00, 01, 10, 00, 01, 10, 00, etc.). The 11 state (after the initial fragment) is skipped in this process to avoid confusion with the single fragment of a non-continued message. The final fragment is indicated by the Message Continued Flag being reset to 0.
N	**Message Number**—When the system supports message retrieval the system controller assigns message numbers (for each paging address separately) starting at 0 and progressing up to a maximum of 63 in consecutive order. The actual maximum roll over number is defined in the pager code plug to accommodate values set in the system infrastructure. When message numbers are not received in order, the subscriber should assume a message has been missed. The subscriber or the pager may determine the missing message number(s) allowing a request to be made for retrieval. When a normal unnumbered numeric message is received (message retrieval flag is equal to 0), it is not to be included in the missed message calculation. This number is also used to identify fragments of the same message. Multiple messages to the same address must have separate message numbers. An exception to this rule is the header message tied to a transparent message, each with the same message number.
R	**Message Retrieval Flag**—When this bit is set to 1, the pager expects to see messages numbered in order (each address numbered separately). Detection of a missing number indicates a missed message. A message received with **R** = 0 is allowed to be out of order and not cause the pager to indicate that a message has been missed.

Table A-6 Hex/Binary Message Bit Definitions (Continued)

Symbol[1]	Definition
M	**1-bit Mail Drop Flag**—When set to 1, this bit indicates the message is to be stored in a special area in memory. It automatically writes over existing data in that memory space.
D	**1-bit Display Direction Field**— • **D** = 0—Display left to right • **D** = 1—Display right to left (valid only when data sent as characters (i.e., Blocking Length not equal 0001)).
H	**1-bit Header Message**— • **H** = 1—Indicates that this message is a header to a following transparent message of the same message number • **H** = 0—Implies message is not a header
B	**4-bit Blocking Length**—This bit field indicates the number of bits per character. • $\mathbf{B_3B_2B_1B_0}$ = 0001—1 bit per character (binary/transparent data) • $\mathbf{B_3B_2B_1B_0}$ = 1111—15 bits per character • $\mathbf{B_3B_2B_1B_0}$ = 0000—16 bits per character Data with blocking length other than 1 is assumed to be displayed on a character by character basis. (default value = 0001)
s	**5-bit Field Reserved for future use**—Default value = 00000
S	**8-bit Signature Field**—The signature is defined to be the one's complement of the binary sum over the total message taken 8 bits at a time prior to formatting into fragments. It would be equivalent to a binary sum starting with the first 8 bits directly following the signature field ($\mathbf{b_3b_2b_1b_0a_3a_2a_1a_0}$ + $\mathbf{d_3d_2d_1d_0c_3c_2c_1c_0}$ and so on) and continuing all the way to the last valid data bit in the last word of the last fragment. The 8 Least Significant Bits of the result are inverted (one's complement) and transmitted as the message signature.[2]

Note: 1. Fields **R** through **S** are only in the first fragment of a message. The fields **K** through **N** make up the first word of every fragment in a long message.
2. This sum does not include any termination bits and should be calculated directly on the message as received by the terminal. The device generating the signature should be able to calculate before the fragmenting boundaries are determined.

MESSAGE CONTENT

Starting with the first character of the third word in the message (second word in the remaining fragments), each 4-bit field represents one of any of the sixteen possible combinations with no restrictions (data may be binary).

FRAGMENT TERMINATION

Unused bits in the last message word of a fragment are filled with all 0s or all 1s, depending on the last valid data bit. This choice is always the opposite polarity of the last valid data bit. For first fragments and inner fragments of a multi-fragment message, the message is interrupted (stopped) on the last full character boundary in the last code word in the fragment. Any unused bits follow the rule just stated. The final fragment follows the above rules except when the last character is all 1s or all 0s and it exactly fills the last code word. In this case, an additional word must be sent of opposite polarity of all 1s or all 0s to signify the position of the last character, thus allowing that last character to be an all 1s or an all 0s character pattern.

Note: This is always the case when a binary message ends in the last bit of the last word.

MESSAGE HEADER

A message header is designated by setting the **H** bit to 1. This is a displayable tag associated with a transparent non-displayable data message. The tag and the associated message are complete in themselves. The pager associates the header message with the data file based on the two having the same message number and being sent in sequence (header first followed by data file).

Alphanumeric Message

The following tables describe the bit format of the alphanumeric messages. The 7-bit characters of the message are designated as lower case letters a, b, c, d, etc. Alphanumeric messages can be sent as fragments. The service provider has the option of dividing the message into several pieces and sending the separate pieces at any time within a given time period.

Table A-7 Vector type V=101 First Only Fragment

Message Word	**i_0**	**i_1**	**i_2**	**i_3**	**i_4**	**i_5**	**i_6**	**i_7**	**i_8**	**i_9**	**i_{10}**	**i_{11}**	**i_{12}**	**i_{13}**	**i_{14}**	**i_{15}**	**i_{16}**	**i_{17}**	**i_{18}**	**i_{19}**	**i_{20}**
1st	K_0	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9	C_0	F_0	F_1	N_0	N_1	N_2	N_3	N_4	N_5	R_0	M_0
2nd	S_0	S_1	S_2	S_3	S_4	S_5	S_6	a_0	a_1	a_2	a_3	a_4	a_5	a_6	b_0	b_1	b_2	b_3	b_4	b_5	b_6
3rd	c_0	c_1	c_2	c_3	c_4	c_5	c_6	d_0	d_1	d_2	d_3	d_4	d_5	d_6	e_0	e_1	e_2	e_3	e_4	e_5	e_6
4th	f_0	f_1	f_2	f_3	f_4	f_5	f_6	g_0	g_1	g_2	g_3	g_4	g_5	g_6	h_0	h_1	h_2	h_3	h_4	h_5	h_6
5th	i_0	i_1	i_2	i_3	i_4	i_5	i_6	j_0	j_1	j_2	j_3	j_4	j_5	j_6	k_0	k_1	k_2	k_3	k_4	k_5	k_6
...																					