Domosys Corporation

# CEWay™ PL-One

CEWay™ is a family of chips developed by DOMOSYS Corporation to meet the requirements of the residential and commercial local area networks (LANs). The CEWay PL-One is the ideal device for simple nodes, such as switches, actuators and sensors. It integrates the complete Physical Layer of the CEBus® standard (EIA-600) and an M8052 core microcontroller. It provides you with all of the resources you need to embed the upper layers of the CEBus standard and the user application into a single-chip solution. The CEWay PL-One is designed for superior performance in noisy power line environments.

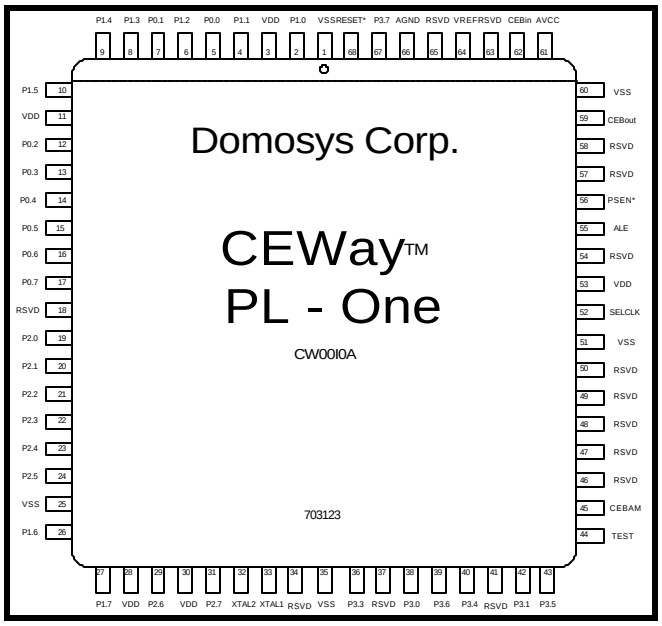



*Figure 1 Pin-out for CEWay PL-One*

## Features

- CEBus power line Physical Layer
- Power line medium dependent Physical and Symbol Encoding Sublayers
- Proprietary DSP for superior signal Reception in noisy environments
- M8052 core microcontroller
- 4 SFRs for communication between PLSES and M8052
- Up to 64 KB of external data memory
- 256 bytes of internal data memory
- Up to 64 KB of external code memory (can be extended with bank switching)
- Three 16-bit timer/counters
- Full-duplex serial port
- 15 I/O pins
- 68-pin PLCC package
- Industrial operating temperature range

## CEWay PL-One Block Diagram

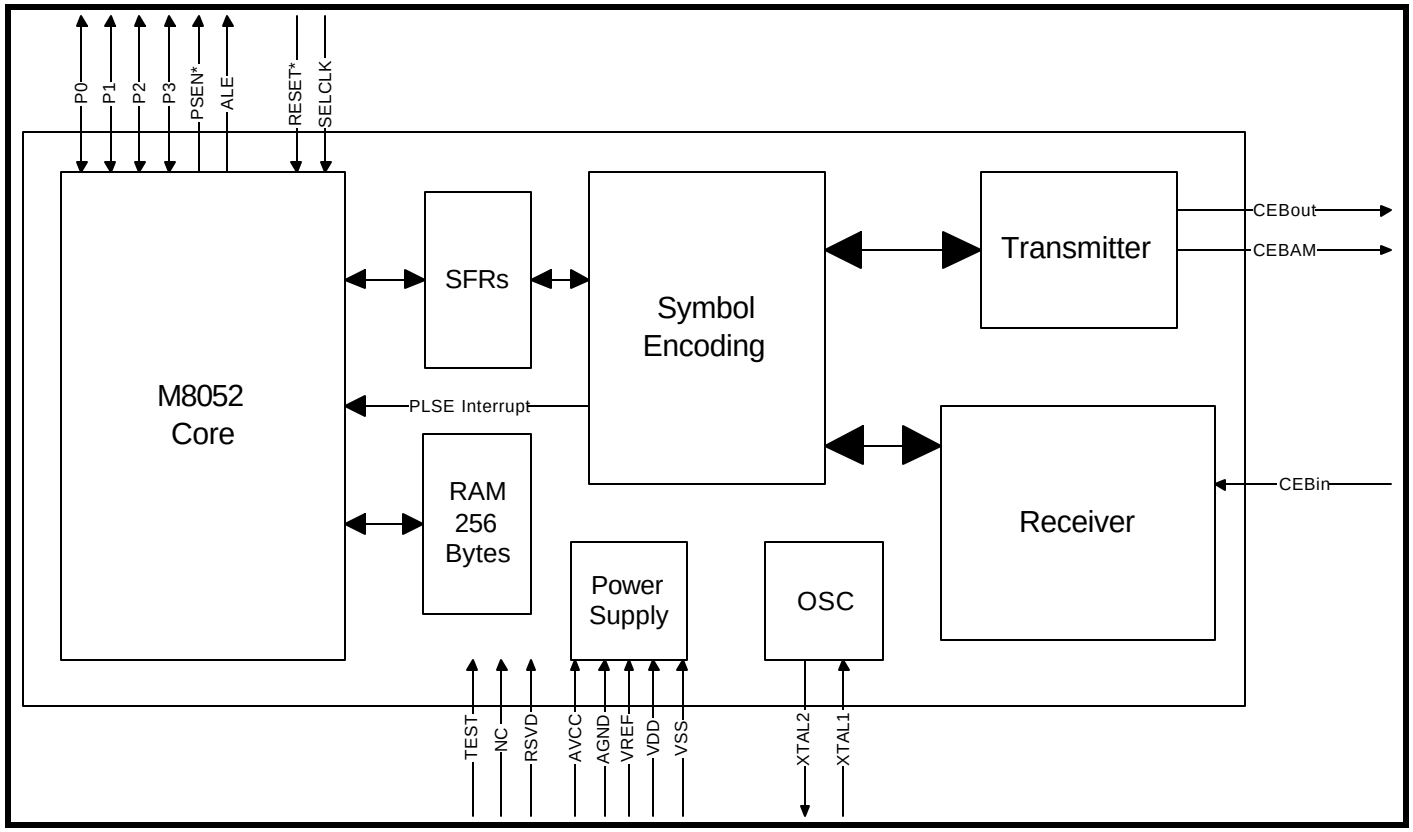



*Figure 2 CEWay PL-One Block Diagram*

## Pin Descriptions

*Table 1 CEWay PL-One Pin Descriptions*

| Symbol | PIN | Description |
|---|---|---|
| P0.0 – P0.7 | 5, 7, 12, 13, 14, 15, 16, 17 | **Port 0:** 8-bit bi-directional I/O port<br>Port Pin — Alternative Function<br>P0.0 — A/D0<br>P0.1 — A/D1<br>P0.2 — A/D2<br>P0.3 — A/D3<br>P0.4 — A/D4<br>P0.5 — A/D5<br>P0.6 — A/D6<br>P0.7 — A/D7 |
| P1.0 – P1.7 | 2, 4, 6, 8, 9, 10, 26, 27 | **Port 1:** 8-bit bi-directional I/O port<br>Port Pin — Alternative Function<br>P1.0 — T2<br>P1.1 — T2EX<br>P1.2 — -<br>P1.3 — -<br>P1.4 — -<br>P1.5 — -<br>P1.6 — -<br>P1.7 — - |
| P2.0 – P2.7 | 19, 20, 21, 22, 23, 24, 29, 31 | **Port 2:** 8-bit bi-directional I/O port<br>Port Pin — Alternative Function<br>P2.0 — A8<br>P2.1 — A9<br>P2.2 — A10<br>P2.3 — A11<br>P2.4 — A12<br>P2.5 — A13<br>P2.6 — A14<br>P2.7 — A15 |
| P3.0, P3.1<br>P3.3 – P3.7 | 38, 42, 36, 40, 43, 39, 67 | **Port 3:** 7-bit I/O port. 2 bi-directional pins, 3 outputs, and 2 inputs. Output pins always read as 0.<br>Port Pin — Alternative Function — Direction<br>P3.0 — RxD — Bi-directional<br>P3.1 — TxD — Bi-directional<br>P3.2 — INT0: Not available externally, used for PLSES Interrupt. — N/A<br>P3.3 — INT1 — Input<br>P3.4 — T0 — Output<br>P3.5 — T1 — Input<br>P3.6 — WR* — Output<br>P3.7 — RD* — Output |
| XTAL2 | 32 | Crystal 2: Output to the inverting oscillator amplifier that forms the oscillator. |

| Symbol | PIN | Description |
|---|---|---|
| XTAL1 | 33 | Crystal 1: Input to the inverting oscillator amplifier that forms the oscillator. |
| CEBAM | 45 | Output: Digital output used to enable and disable the transmit amplifier. |
| CEBOUT | 59 | CEBus power line signal output |
| CEBIN | 62 | CEBus power line signal input |
| SELCLK | 52 | SELCLK: Must be connected to VDD if 14.31818 MHz is used. Connected to VSS if 21.4772 MHz is used. |
| PSEN* | 56 | PSEN*: Reads strobe to the external program memory via Port 0 and Port2. |
| ALE | 55 | ALE: Latches the low byte of the address during access of external memory in normal operation. |
| VREF | 64 | Input for analog reference. A 1 µF capacitance must be put between this pin and AGND. |
| RESET* | 68 | Reset: A LOW on this pin for 24 machine cycles and while the oscillator is running resets the device. This pin has an internal pull-up resistor. Note that, unlike most 8051 devices, no internal Schmitt Trigger is present on this pin. |
| AGND | 66 | Ground: 0 volt Analog Reference |
| VDD | 3, 11, 28, 30, 53 | Power Supply: Digital 5 volts |
| VSS | 1, 25, 35, 51,60 | Ground: 0 volt Digital Reference |
| AVCC | 61 | Power Supply: Analog 5 volts |
| RSVD | 18, 34, 37, 41, 46, 47, 48, 49, 50, 54, 57, 58, 63, 65 | Reserved - Must be connected to VSS. |
| TEST | 44 | TEST: Must be connected to VSS. |

# Electrical Specifications

### Absolute Maximum Rating[1]

| Parameter | Sym | Min | Max | Units | Test Conditions |
|---|---|---|---|---|---|
| Supply Voltage | $V_{DD}$- $V_{ss}$ | -0.3 | 7 | V | |
| DC Input Voltage | $V_{IN}$ | -0.3 | VDD +0.3 | V | |
| DC Input Current | $I_{IN}$ | -10 | +10 | mA | |
| Storage Temperature | $T_{STG}$ | -40 | +125 | $^{O}$C | |
| ESD Tolerance | | | 2 | kV | |

[1] Exceeding these values may cause permanent damage. Functional operation under these conditions is not implied.

**DC Electrical Characteristics -** Voltages are with respect to VSS unless otherwise stated

| Parameter | Sym | Min | Typ[2-3] | Max | Units | Test Conditions |
|---|---|---|---|---|---|---|
| Supply Voltage - Digital | VDD | 4.75 | 5 | 5.25 | V | |
| Supply Voltage - Analog | AVCC | 4.75 | 5 | 5.25 | V | |
| Input Voltage (high) | $V_{IH}$ | 0.7VDD | | VDD+0.3 | V | |
| Input Voltage (low) | $V_{IL}$ | VSS-0.3 | | 0.3VDD | V | |
| Output Voltage (high) | $V_{OH}$ | 2.4 | | | V | $I_{OH}$ = 50 µA |
| Output Voltage (low) | $V_{OL}$ | | | 0.4 | V | $I_{OL}$ = 4 mA |
| Operating Current Digital | $I_{VDD}$ | | 25 | 35 | mA | $f_{CLK}$ = 14.318 MHz |
| Operating Current Analog | $I_{AVCC}$ | | 9.5 | 15.5 | mA | $f_{CLK}$ = 14.318 MHz |
| Operating Temperature | $T_O$ | -40 | | +85 | $^{O}$C | |

[2] Typical figures are at 25 OC and are for design aid only: not guaranteed and not subject to production testing.

[3] DC Electrical Characteristics are over recommended temperature range & recommended power supply voltages.

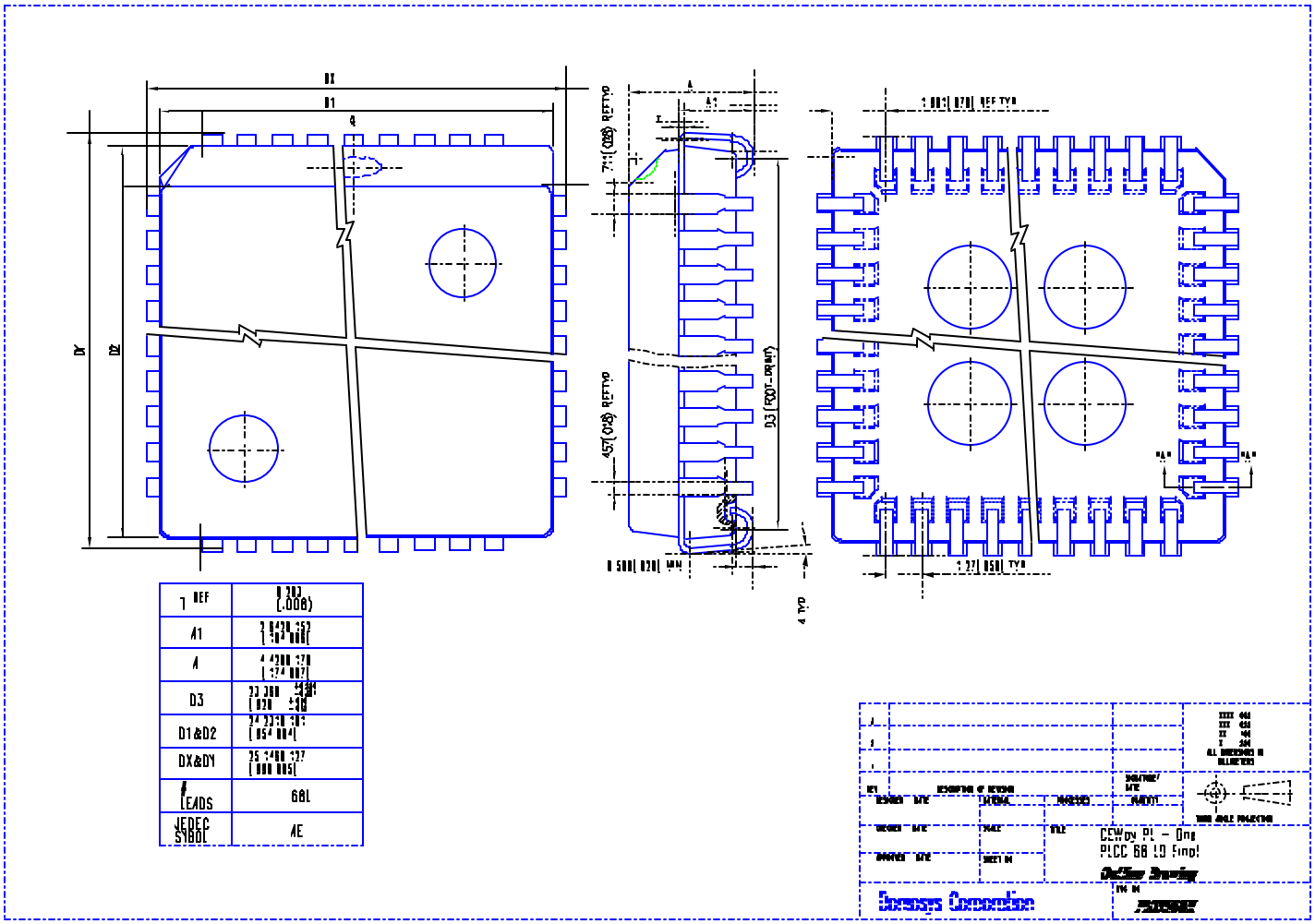
### AC Electrical Characteristics

| Parameter | Sym | Min | Typ[4] | Max | Units | Test Conditions |
|---|---|---|---|---|---|---|
| CEBout Output Voltage | $V_{CEBout}$ | | 3.5 | | V p-p | |
| CEBout Load Impedance | $ZL_{CEBout}$ | 5 | | | kΩ | |
| CEBin Input Impedance | $Zin_{CEBin}$ | 40 | 68 | 140 | kΩ | |
| Clock Frequency[5] | $f_{CLK}$ | 14.31818 | | 21.4772 | MHz | |
| Pin Capacitance XTAL1,XTAL2 | $C_{XTAL}$ | | 0.8 | | pF | |
| Input Pin Capacitance | $C_I$ | | 8 | | pF | $f_{CLK}$ = 14.318 MHz |
| Output Pin Capacitance | $C_o$ | | 8 | | pF | $f_{CLK}$ = 14.318 MHz |

[4] Typical figures are at 25 OC and are for design aid only: not guaranteed and not subject to production testing.

[5] Clock frequency can only be either 14.31818 or 21.4772 MHz for proper chirp timing.

## Packaging



## Ordering Information

| Part Number | Description |
|---|---|
| PONE-000-01 | CEWay PL-One Integrated Circuit |

## CEBus Standard Implementation

The CEWay PL-One fully implements the CEBus Standard power line Physical Layer (PhyL) using dedicated circuitry. The Physical Layer is composed of a Medium Dependant Physical Sublayer (MDPS) and of a Power Line Symbol Encoding Sublayer (PLSES). It also provides an M8052 core microcontroller on-chip with which the upper layers (DLL, NL, and AL) of the CEBus standard or Home Plug & Play specification can be run. Libraries that implement these upper layers are available in CEBox™, the DOMOSYS' software system.

### Medium Dependent Physical Sublayer & Power Line Symbol Encoding Sublayer

The transceiver (MDPS and PLSES) is composed of a transmit chain and a receive chain. The transmitter provides an analog representation of a stored ROM code to an off-chip amplifier for transmission. Upon reception, the device samples the incoming filtered waveform through an ADC and provides a bit-by-bit comparison with an internally stored set of waveform probabilities. When a digital threshold is exceeded, the bit is stored with the rest of the message.

The symbol decoder performs leading zero suppression and recognizes the Preamble EOF (PEOF) string in order to force the device into the receive state. The symbol decoder recognizes the incoming signals from the transversal filter and prepares them for storage in RAM.

The MDPS and PLSES embed a part of the Layer System Management (LSM).

Detailed information on the CEBus Power Line Physical Layer can be found in the following documents:

- EIA 600.31
- IS 60.03 parts 7 & 8

## M8052 to Physical Layer Interface

Even though the Physical Layer is also composed of the MDPS, the PL-One firmware only communicates with the PLSES through 4 Special Function Registers (SFRs). The PLSES then communicates with the MDPS if need be.

### Power Line Symbol Encoding Sublayer (PLSES)

The main task of the PLSES is to encode/decode a sequence of symbols into/from a stream of states. In addition, it has the task of recognizing the beginning of an incoming frame, and preventing a frame collision when the PL-One firmware requests the channel for a transmission. Sometimes it must suppress leading zeros in a given field before transmitting it, or restore them when receiving it. It also computes the CRC checksum appended to the end of a frame while transmitting or receiving it to ensure communication integrity. The PLSES will also detect Jabber conditions, defined as the transmission or reception of 1,000 consecutive SUPERIOR states on the medium, and indicates this via a PLSES Handshake Interrupt. A timestamp can also be provided following any reception or transmission.

The PLSES operates in four main states: Initialization (INIT), Reception (RCV), Transmission (XMIT) and Idle (IDLE). The RCV and XMIT states can be divided into sub-states as shown in Figure 3 State Transition Diagram for PLSES.

The data travel between the PLSES and the M8052 through four Special Function Registers (SFRs). Each one is 8-bit wide. They are denoted: Ph_Confirm_Reg, Ph_Request_Reg, Ph_Rx_Buffer and Ph_Tx_Buffer. When a communication is requested by the PLSES to the PL-One firmware, an interrupt (INT0) is sent to the M8052. The latter then reads the flags in the Ph_Confirm_Reg SFR to determine the type of interrupt service routine to execute. The description of all flags is shown in Table 3. On the other hand, when the PL-One firmware requests an action by the PLSES, the former writes a control value to the Ph_Request_Reg. The description and the configuration of each of the control bits is shown in Table 6.

Data travels from the M8052 to the PLSES through the Ph_Tx_Buffer SFR one byte at a time. Similarly, the data is read by the PL-One firmware through the Ph_Rx_Buffer SFR one byte at a time.

#### Ph_Rx_Buffer

It is used to pass on the packet Data symbols. It should be pointed out that when the Leading Zero Suppression function has been carried out, the empty positions are padded with zeros. It is also used to pass up both the reception and transmission timestamp bytes.

#### Ph_Confirm_Reg

The values of the 6 LSBits of this port are valid for exactly one read cycle after receiving a PLSES Handshake Interrupt. When this port has been read once, the LSBits' values will not necessarily be valid until the next time a PLSES Handshake Interrupt is received. The 2 MSBits (*ch_active* and *ch_noisy*) of this port are always valid.

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Ph_Confirm_Reg |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | 0 | *jabber* |
| | | | | | | 0 | | *rx_del$_0$* |
| | | | | | 0 | | | *rx_del$_1$* |
| | | | | 0 | | | | *gd/packet* |
| | | | 0 | | | | | *be/bf* |
| | | 0 | | | | | | *col/rx* |
| | 0 | | | | | | | *ch_noisy* |
| 0 | | | | | | | | *ch_active* |

*Table 2 Ph_Confirm_Reg in Idle State*

| **Bits in Ph_Confirm_Reg** | |
|---|---|
| **Bit** | **Description** |
| *ch_active (MSB)* | Description: Channel active indicates the presence of a CEBus-like signal on the channel.<br>Hardware (HW)Reset state: 0<br>Enable trigger: Set to 1 whenever a chirp is detected on the medium. It is maintained as long as chirps continue to be detected. This includes the special behavior described for the Ph_Confirm_Reg *col/rx* flag. It is also maintained while PLSES receives one symbol and is trying to identify a symbol noise condition as described for the Ph_Confirm_Reg *ch_noisy* flag.<br>Reset trigger: It is reset as soon as the reception of a packet ends (after the reception of the CRC or after a Carrier Lost), when already within the RCV state. If in the IDLE state, it means it was activated from a Symbol Noise. It should then be reset as soon as *ch_noisy* is enabled. |
| *ch_noisy* | Description: Channel noisy. In IDLE state, it indicates the presence of an isolated strong peak on the channel (Symbol Noise). At the transition from IDLE to the RCV state, it indicates a frame buried in noise (Medium Noisy). During the rest of the RCV state, it indicates the loss of the carrier (Carrier Lost).<br>HW Reset state: 0<br>Enable trigger: Symbol Noise: Set to 1 when one SUPERIOR Preamble symbol, or 1 PEOF delimiter symbol, followed by 5 consecutive INFERIOR Preamble symbols (1.14 USTs) have been detected. As soon as the 5$^{th}$ consecutive symbol is identified as an INFERIOR, *ch_noisy* is set to 1. Medium Noisy: It is also set to 1 when only 3 to 6 (inclusive) PEOF symbols were detected. Carrier Lost: It is set to 1 when, after having received at least a PEOF and any other symbols after that PEOF, and before a complete CRC has been received, an INFERIOR state is detected on the medium.<br>Reset trigger: In a Symbol Noise condition, it is reset as soon as a new chirp is detected. If no new chirps are detected, or for the two other conditions, it is reset when the PLSES Handshake Interrupt is reset. |

| Bits in Ph_Confirm_Reg | |
|---|---|
| **Bit** | **Description** |
| *col/rx* | Description: Collision/Reception. In the XMIT state, it indicates the detection of a collision with another frame during the transmission of the Preamble field. In the RCV state, it indicates that you are past the Preamble and PEOF and into the body of a packet.<br>HW Reset state: 0<br>Enable trigger: Set to 1 when a collision is detected during transmission of the Preamble (when the PLSES transmits an INFERIOR but detects a SUPERIOR state on the medium). Also set to 1 as soon as the PLSES has detected a PEOF in reception, and remains at 1 throughout the reception. It remains enabled for a Carrier Lost indication.<br>Reset trigger: It is reset when the PLSES Handshake Interrupt is reset.<br>Special constraints: Following a collision the PLSES will automatically transition from its XMIT state to its RCV state. At the same time, *ch_active* is enabled since a SUPERIOR Preamble symbol has been detected over our INFERIOR Preamble symbol. |
| *be/bf* | Description: Buffer Empty/Buffer Full. In XMIT state, it indicates that the Ph_Tx_Buffer has been transmitted successfully and is empty. This flag is also set to 1 when the CRC checksum has been appended to the end of the frame and transmitted. In RCV state, it indicates that the data within the Ph_Rx_Buffer is valid.<br>HW Reset state: 0<br>Enable trigger: Set to 1 after the PLSES has transmitted on the medium a previous packet field part, via a request with the Ph_Request_Reg. Set to 1 when the PLSES has stored packet Preamble symbols or Data symbols in the Ph_Rx_Buffer.<br>Reset trigger: It is reset when the PLSES Handshake Interrupt is reset. |
| *gd_packet* | Description: Good Packet. In RCV state, it indicates that no error has been detected in the CRC checksum. If it is not set after having received the CRC, it means that the incoming packet was corrupted during the communication.<br>HW Reset state: 0<br>Enable trigger: Set to 1 after the last CRC symbol has been received and the CRC checksum has been verified as successful.<br>Reset trigger: It is reset when the PLSES Handshake Interrupt is reset. |

| Bits in Ph_Confirm_Reg | |
|---|---|
| **Bit** | **Description** |
| *rx_del*$_1$<br>*rx_del*$_0$ | Description: Reception delimiters. In RCV state, this pair of bits indicates the type of delimiter which terminates the current byte. Their configuration is shown below in Table 4.<br>HW Reset state: 00<br>Enable trigger:<br>• Set to 11 when a PEOF symbol has been detected and received.<br>• Set to 10 when an EOP symbol has been detected and received. Also set to 10 after the CRC has been received and the checksum result is available, or set to 10 when the reception timestamp bytes are being passed to the PL-One firmware.<br>• Set to 01 when an EOF symbol has been detected and received.<br>• Set to 00 when 8 consecutive packet Data symbols have been saved in Ph_Rx_Buffer before receiving an EOF or EOP delimiter symbol. This case can happen for any packet field part, including those transmitted with Leading Zero Suppression (LZS). Also set to 00 when a PEOF symbol has been received but fewer than 8 Preamble symbols were detected. This situation is likely to happen with a Medium Noisy indication (see *ch_noisy* bit).<br>Reset trigger: It is reset when the PLSES Handshake Interrupt is reset. |
| *jabber (LSB)* | Description: Jabber Detect. In XMIT state, it indicates that the PL-One has transmitted at least 1,000 consecutive SUPERIOR symbols. In RCV state, it indicates that the PL-One has received at least 1,000 consecutive SUPERIOR symbols.<br>HW Reset state: 0<br>Enable trigger: Set to 1 after the PL-One has transmitted at least 1,000 consecutive SUPERIOR symbols. Also set to 1 after the PL-One has received at least 1,000 consecutive SUPERIOR symbols.<br>Reset trigger: It is reset when 1024 consecutive SUPERIOR states have been detected on the channel in either transmission or reception. |

*Table 3 Explanation of Bits in Ph_Confirm_Reg SFR*

| Symbol | *del*$_1$ | *del*$_0$ | # USTs |
|---|---|---|---|
| EOF | 0 | 1 | 3 |
| EOP | 1 | 0 | 4 |
| PEOF | 1 | 1 | 8 |
| No delimiter | 0 | 0 | 0 |

*Table 4 Bit Representation of Four Possible Delimiters (EOF, EOP, PEOF, and none)*

## PLSES Handshake Interrupt Generation

The PL-One firmware is supplied with an interrupt from the PLSES which is called the "PLSES Handshake Interrupt". This uses the standard 8052 INT0* interrupt. This means that the port bit P3.2 is no longer available for the user application. P3.2 will normally read as 1, but will fall to 0 as soon as an interrupt is generated from the PLSES. It will remain at 0 until the Ph_Confirm_Reg is read.

Note that this interrupt (INT0) MUST be set to the high priority level to ensure that no packets are lost. For this interrupt to be properly configured, the following four bits need to be set:

- PX0: Sets INT0 to high priority
- IT0: INT0 triggers on a falling edge
- EX0: INT0 is enabled
- EA: Enable interrupts

The PLSES Handshake Interrupt is used by the PL-One firmware to perform its communication handshake with the PLSES, and may indicate that valid data is present in the Ph_Rx_Buffer. It is provided by PLSES based on some specific PLSES states and events. These are defined in Table 7. The Ph_Confirm_Reg needs to always be read to determine the cause of the interrupt, and to process it accordingly.

The PLSES Handshake Interrupt is reset only after a PLSES Handshake Interrupt has been generated, followed by the reading of the Ph_Confirm_Reg. Note that the PLSES Handshake Interrupt needs to have been generated first, in order to distinguish between the Ph_Confirm_Reg being read by the user application while in its main processing. This can be shown as follows:

1. PLSES Handshake Interrupt is generated
2. Ph_Confirm_Reg is read
3. The PLSES Handshake Interrupt is reset

### Ph_Tx_Buffer

The Ph_Tx_Buffer is used to pass initialization parameters and data to be transmitted down to the PLSES. Setting the *lzs* bit in Ph_Request_Reg will result in all leading zeros in this byte not being transmitted. To simply transmit a field delimiter without any data symbols, this byte should be loaded with 00 hex, and transmitted with leading zero suppression. The only exception to this is when transmitting the Preamble field, in which case the contents of the Ph_Tx_Buffer have no effect on the Preamble which is transmitted.

### Ph_Request_Reg

The Ph_Request_Reg is initialized with 00 hex following a hardware reset. The Ph_Request_Reg is read and interpreted by the PLSES each time it is written to. This value is maintained until it is next written to by the PL-One firmware. The various state transitions that can be brought about by writing to Ph_Request_Reg are listed in Table 8.

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Ph_Request_Reg |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | 0 | *stamp* |
| | | | | | | 0 | | *tx_del$_0$* |
| | | | | | 0 | | | *tx_del$_1$* |
| | | | | 0 | | | | *reserved* |
| | | | 0 | | | | | *Izs* |
| | | 0 | | | | | | *stop_rx* |
| | 0 | | | | | | | *hw_reset* |
| 0 | | | | | | | | *tx* |

*Table 5 Value to write to Ph_Request_Reg SFR to stop transmission and return to Idle state*

| **Bits in Ph_Request_Reg** | |
|---|---|
| **Bit** | **Description** |
| *tx* (MSB) | Transmission. In IDLE state, it forces the PLSES to enter the XMIT state. In XMIT state, it should always be high. In RCV state, it should always be low. |
| *hw_reset* | Hardware Reset. In all states, this bit forces the whole Physical Layer (PhyL) to perform a hardware reset and return to the INIT state. |
| *stop_rx* | Stop reception. In the INIT state, it is set when passing initialization parameters to the PLSE. In the RCV state, it forces the PLSES to stop reception and enter the IDLE state. |
| *lzs* | Leading Zero Suppression. In the INIT state, it is set when passing initialization parameters to the PLSE. In the XMIT state, it forces the PLSES to suppress the leading zeros in the Ph_Tx_Buffer. When transitioning from the IDLE to the XMIT_PRE_SYM state, it forces the PLSES to suppress the Preamble symbols; in this case, only the PEOF is transmitted (e.g. for transmitting an IACK or IRetry packets). |
| *reserved* | Reserved |
| *tx_del$_1$*<br>*tx_del$_0$* | Transmission delimiters. This pair of bits is used to select the type of delimiter to transmit after the byte contained in Ph_Tx_Buffer. The mapping between these bits and the possible delimiters is shown in Table 4. |
| *stamp* (LSB) | Stamp. At the end of the RCV or XMIT states, this flag indicates to the PLSES that the PL-One firmware needs one more significant timestamp byte (to a maximum of four total) recorded when the last CRC symbol was received or transmitted. |

*Table 6 Explanation of Bits in the Ph_Request_Reg SFR*

### State Transition Tables for Ph_Confirm_Reg and Ph_Request_Reg

The following two tables (Table 7 and Table 8) show the state transitions associated with the PLSES Handshake Interrupts (reads of Ph_Confirm_Reg), and writes to Ph_Request_Reg.

| Current State | Next State | Ph_Confirm_Reg | Transition |
|---|---|---|---|
| IDLE | RCV_PRE_SYM | 1000 0000 | Detection of first SUPERIOR and *stop_rx* bit not active. |
| XMIT_PRE_SYM | XMIT_PRE_SYM | 0001 0000 | After command to start transmission. |
| IDLE | IDLE | 0000 0000 | Ch_Quiet indication every UST. |

| Current State | Next State | Ph_Confirm _Reg | Transition |
|---|---|---|---|
| RCV_PRE_SYM | RCV_PRE_SYM | 1000 0000 | Detection of INFERIOR or SUPERIOR state and<br>1) if current state is INFERIOR, the number of consecutive INFERIORs is < 5, or<br>2) if current state is SUPERIOR, it is of the same phase as the first SUPERIOR. |
| RCV_PRE_SYM | IDLE | 0100 0000 | Detection of 5 consecutive INFERIORs. |
| RCV_PRE_SYM | RCV_SYM | 1011 0110 | Detection of opposite phase SUPERIOR after previously detecting 7 or 8 consecutive SUPERIORs. |
| RCV_PRE_SYM | RCV_SYM | 1111 0110 | Detection of opposite phase SUPERIOR after previously detecting between 3 and 6 (inclusive) consecutive SUPERIORs. |
| RCV_PRE_SYM | RCV_PRE_SYM | XXXX XXX1 | Jabber error. |
| RCV_SYM | RCV_CRC | 1011 0100 | Detection of EOP and valid data in Ph_Rx_Buffer. |
| RCV_SYM | RCV_CRC | 1010 0100 | Detection of EOP and no data in Ph_Rx_Buffer. |
| RCV_SYM | RCV_SYM | 1011 0000 | Detection of data and 8 valid data bits in Ph_Rx_Buffer. |
| RCV_SYM | RCV_SYM | 1011 0010 | Detection of EOF and valid data in Ph_Rx_Buffer. |
| RCV_SYM | RCV_SYM | 1010 0010 | Detection of EOF and no data in Ph_Rx_Buffer. |
| RCV_SYM | RCV_SYM | 0110 0000 | Detection of INFERIOR, as a result of carrier loss. |
| RCV_SYM | RCV_SYM | XXXX XXX1 | Jabber error. |
| RCV_CRC | RCV_STAMP | 1011 X100 | Received 16 CRC bits to report CRC status. |
| RCV_CRC | RCV_CRC | 0110 0000 | Detection of INFERIOR prior to all 16 CRC bits, as a result of carrier loss. |
| RCV_CRC | RCV_CRC | XXXX XXX1 | Jabber error. |
| RCV_STAMP | RCV_STAMP | 1011 0100 | After each timestamp byte is uploaded. |
| XMIT_PRE_SYM | RCV_PRE_SYM | 1010 0000 | Collision detected. |
| XMIT_PRE_SYM | XMIT_SYM | 0001 0000 | Transmission of Preamble completed. |
| XMIT_PRE_SYM | XMIT_PRE_SYM | XXXX XXX1 | Jabber error. |
| XMIT_SYM | XMIT_SYM | 0001 0000 | Buffer empty. |
| XMIT_SYM | XMIT_CRC | | Start transmission of CRC. No interrupt is generated. |
| XMIT_SYM | XMIT_SYM | XXXX XXX1 | Jabber error. |
| XMIT_CRC | XMIT_STAMP | 0001 0000 | Transmission of CRC completed. |
| XMIT_CRC | XMIT_CRC | XXXX XXX1 | Jabber error. |
| XMIT_STAMP | XMIT_STAMP | 0001 0000 | After each timestamp byte is uploaded. |

*Table 7 State Transitions Associated with PLSES Handshake Interrupts*

| Current State | Ph_Request _Reg | Next State | Transition |
|---|---|---|---|
| IDLE | 101x 0110 | XMIT_PRE_SYM | Command to start transmission. |
| RCV_PRE_SYM | X01X XXXX | IDLE | Abort reception. |
| RCV_SYM | X01X XXXX | IDLE | Abort reception. |
| RCV_CRC | X01X XXXX | IDLE | Abort reception. |
| RCV_STAMP | X01X XXXX | IDLE | Abort reception. |
| RCV_STAMP | 0000 0001 | RCV_STAMP | This indicates that the MACS needs one more timestamp byte. |
| XMIT_PRE_SYM | 101X 0XX0 | XMIT_SYM | Bits 4, 2, and 1 indicate leading zero suppression and field delimiter. |
| XMIT_PRE_SYM | 00XX XXXX | IDLE | Abort transmission, |
| XMIT_SYM | 101X 0XX0 | XMIT_SYM | Bits 4, 2, and 1 indicate leading zero suppression and field delimiter. |
| XMIT_SYM | 00XX XXXX | IDLE | Abort transmission. |
| XMIT_CRC | 00XX XXXX | IDLE | Abort transmission. |
| XMIT_STAMP | 0000 0000 | IDLE | Complete transmission, does not require any more timestamp bytes. |
| XMIT_STAMP | 1010 0001 | XMIT_STAMP | This indicates that the MACS needs one more timestamp byte. |

*Table 8 State Transitions Associated with Writes to Ph_Request_Reg*

**Time stamp Timer**

The PLSES supports a free-running 32-bit timer that is used to associate a timestamp to any received or transmitted packets. The resolution of this timestamp timer is 279 nsec (four oscillator cycles at 14.32 MHz, six cycles at 21.48 MHz.) For a received packet, the timestamp is acquired when the last CRC symbol has been completely received. For a transmitted packet, the timestamp is acquired when the CRC symbol has been completely transmitted. The least significant timestamp byte is transferred first, via Ph_Rx_Buffer, at the same time as the end of transmission or reception is indicated. Up to three additional timestamp bytes can be requested from the PLSES by writing 01 hex to the Ph_Request_Reg. These bytes are transferred with the least significant byte first.

## PLSES - PL-One firmware Interface State Diagram

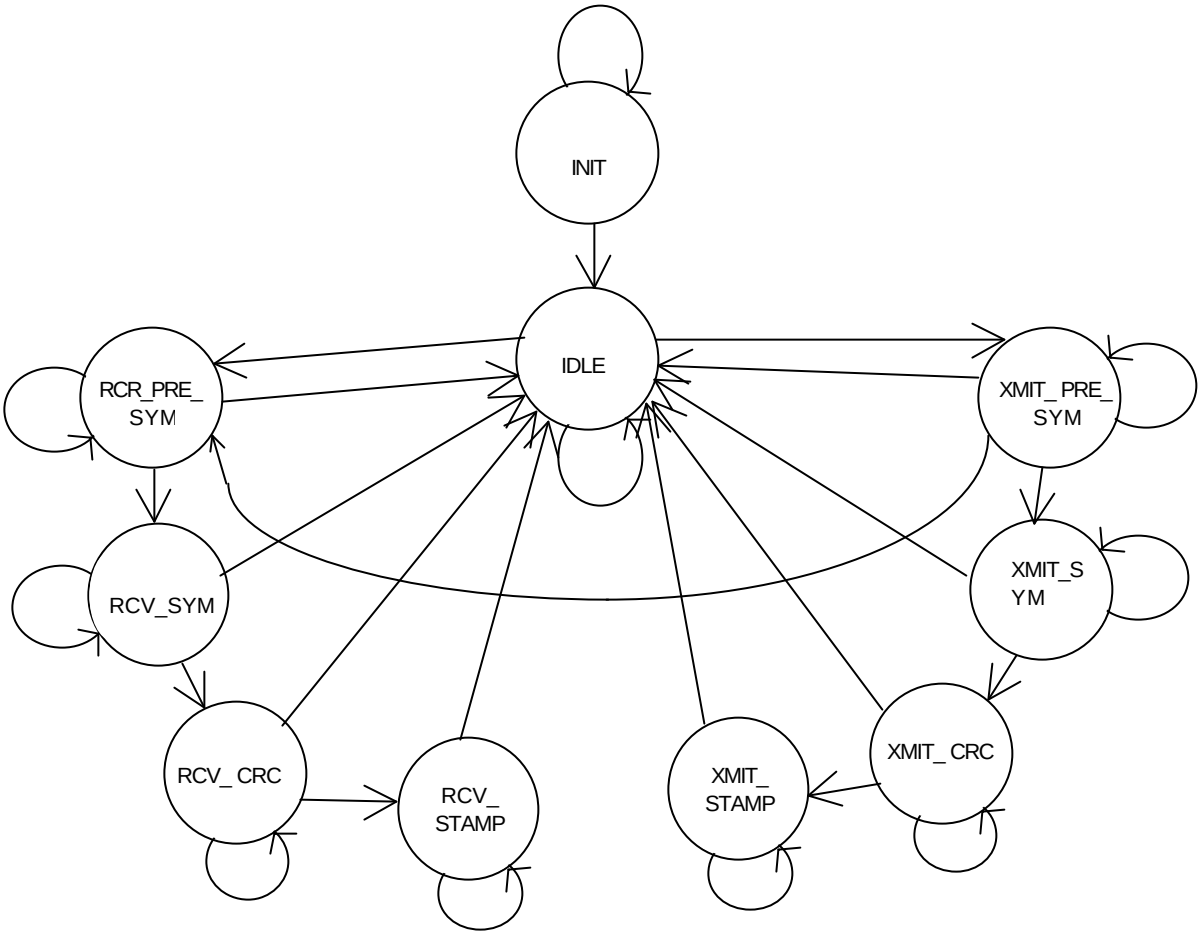



*Figure 3 State Transition Diagram for PLSES*

Note: Setting the Ph_Request_Reg *hw_reset* bit will result in a transition to the INIT state from any other state or sub-state. This is not indicated on this diagram in order to simplify it.

## PLSES State Transition Table

| State | Description |
|---|---|
| INIT State | The SES block goes into INIT state only after a hardware reset, or after the PL-One firmware sets the Ph_Request_Reg *hw_reset* bit. Two steps are required for the PL-One firmware to fully initialize the PLSES and get it to the IDLE state.<br>1. The PLSE registers must be initialized by writing the following sequence of bytes:<br>Write DC hex to Ph_Tx_Buffer<br>Write 30 hex to Ph_Request_Reg<br>Write DC hex to Ph_Tx_Buffer<br>Write 30 hex to Ph_Request_Reg<br>Write 14 hex to Ph_Tx_Buffer<br>Write 30 hex to Ph_Request_Reg<br>Write 20 hex to Ph_Request_Reg<br>Write 00 hex to Ph_Request_Reg<br>2. Second, an initial Preamble value must be written to Ph_Tx_Buffer, which is used as-is for the first transmitted packet. It is also used as a seed for the generation of pseudo-random Preamble values for subsequent transmitted frames.<br>After these two steps, the SES finishes its initialization routine and goes into the IDLE state. |
| IDLE State | In this state the SES has been initialized, is not transmitting a frame, and the channel is quiet. For each unit symbol time (UST) that passes, the SES block will send a PLSES Handshake Interrupt to the PL-One firmware, indicating that there is no activity on the channel. |
| Transition | The SES block will go from IDLE to XMIT_PRE_SYM when the command to transmit a PEOF is written to Ph_Request_Reg (101X 0110 bin). |
| Transition | The SES block will go from IDLE to RCV_PRE_SYM when a SUPERIOR state has been detected on the channel. |

| State | Description |
|---|---|
| RCV_PRE_SYM State | The PLSE Sublayer is in the process of receiving Preamble symbols, which are encoded using Preamble USTs that are 114 microseconds in length. This state counts the number of USTs in consecutive INFERIOR or SUPERIOR states to decode each symbol. At each received medium state transition (a SUPERIOR when LAST_RCV is INFERIOR or an INFERIOR when LAST_RCV is SUPERIOR), the symbol is decoded as "1" or "0" according to how many USTs elapsed between state transitions. The state also waits for the Preamble EOF (PEOF), which is 8 normal USTs in length (100 µs each.) In the case where there are no Preamble symbols, e.g. an IACK frame, the RCV_PRE_SYM state will only process and report the PEOF, and the Ph_Confirm_Reg *bf/be* bit will not be set. |
| Transition | The SES block will go from RCV_PRE_SYM to RCV_SYM after it receives 3 or more SUPERIOR 1s with 100-microsecond (1 UST) spacing, and the medium state changes from SUPERIOR 1 to SUPERIOR 2. |
| Transition | The SES block will go back to IDLE if it detects that the received SUPERIOR state was noise (a SUPERIOR state followed by five INFERIOR states), or if the Ph_Request_Reg *stop_rx* bit is set |
| RCV_SYM State | The PLSE Sublayer is in the process of receiving symbols within the body of the frame. At each received medium state transition (SUPERIOR 1 to SUPERIOR 2, or vice versa), the symbol is decoded according to the elapsed time since the previous transition. The symbol is stored in an 8-bit shift register in the SES block. When the shift register is full, or an EOF (End of Field) or EOP (End of Packet) symbol is received, the SES block will copy the data in the shift register to Ph_Rx_Buffer, and send a PLSES Handshake Interrupt to the PL-One firmware. |
| Transition | The SES will go to RCV_CRC after it receives an EOP symbol. |
| Transition | The SES will go to IDLE when the carrier is lost or when the Ph_Request_Reg *stop_rx* bit is set. The carrier lost condition is explained in a later section. |
| RCV_CRC State | The PLSE Sublayer is in the process of receiving CRC symbols. A bit is shifted into the 16-bit CRC register for CRC computation every UST. A received SUPERIOR 1 shifts in a 1, and a received SUPERIOR 2 shifts in a 0. After 16 USTs, the Ph_Confirm_Reg *gd_packet* bit is set according to whether or not the frame has kept its integrity during reception, and the SES will send a PLSES Handshake Interrupt to the PL-One firmware. |

| State | Description |
|---|---|
| Transition | The SES will also start providing the timestamp to PL-One firmware after the CRC computation, and go to RCV_STAMP. |
| Transition | The SES will go back to IDLE when the carrier is lost or when the Ph_Request_Reg *stop_rx* bit is set. |
| RCV_STAMP State | The PLSE Sublayer is in the process of providing the timestamp to the PL-One firmware. The timestamp is the time associated with the end of the packet CRC reception. The timestamp will be provided on four bytes (LSB first followed by more significant bytes) via the Ph_Rx_Buffer, where each byte will be sent to the PL-One firmware via a PLSES Handshake Interrupt in Ph_Rx_Buffer, and the Ph_Confirm_Reg *be/bf* bit will be set. |
| Transition | The SES will go back to IDLE when the Ph_Request_Reg *stamp* bit (bit 0) is cleared, and the *stop_rx* bit is set. |
| XMIT_PRE_SYM State | The PLSE Sublayer is in the process of transmitting Preamble symbols. |
| Decision | If it is transmitting a long Preamble field (the Ph_Request_Reg *lzs* bit is 0), the SES will first transmit an 8-bit Preamble value. The PL-One firmware loads the initial Preamble value when the SES is in the INIT state. Later it is computed inside the SES through a pseudo-random process. The first symbol transmitted following a quiet channel is always encoded in SUPERIOR state, the next symbol will be encoded in INFERIOR state, the third symbol will be encoded as the same SUPERIOR state as the first symbol, etc. At the completion of transmitting 8-bit Preamble value, the SES will also transmit the PEOF (the same SUPERIOR state for 8 USTs) before going to the XMIT_SYM state. If it is transmitting a short Preamble field (the Ph_Request_Reg *lzs* bit is 1), the SES will not transmit an 8-bit Preamble value, instead it will transmit the 8 UST PEOF directly. But the SES still continues the process for computing the pseudo-random Preamble byte for later use. The SUPERIOR state used by XMIT_PRE_SYM state alternates between SUPERIOR 1 and SUPERIOR 2 for consecutive packets. The first packet transmitted will start with a SUPERIOR 1 state, the second packet transmitted will start with a SUPERIOR 2 state, the third one will be a SUPERIOR 1 and so on. |
| Decision | If it is transmitting a short Preamble field (the Ph_Request_Reg *lzs* bit is 1), no Preamble symbols are transmitted before the 8 SUPERIOR USTs for the PEOF symbol. |
| Action | Before and during the process of transmitting INFERIOR Preamble symbols, the SES checks whether or not the channel is active. If activity is detected on the channel, the SES will send an interrupt to the PL-One firmware indicating a collision and at the same time stop transmission and go to RCV_PRE_SYM. |
| Transition | The SES will go to XMIT_SYM state after it sends Preamble EOF symbol. |

| State | Description |
|---|---|
| Transition | The SES will go back to IDLE if the Ph_Request_Reg *tx* bit is cleared. |
| XMIT_SYM State | The PLSE Sublayer is in the process of transmitting symbols. The symbols are received from the PL-One firmware, one byte at a time through Ph_Tx_Buffer. Starting from LSB, the symbols are encoded as transitions between the SUPERIOR 1 and SUPERIOR 2 states. When the requested medium state lasts 1 UST, a "1" will be shifted into the CRC computation register. When the requested medium state lasts 2 USTs, a "0" will be shifted into the CRC computation register.<br><br>If the Ph_Request_Reg *lzs* bit was set, then none of the leading zeros in Ph_Tx_Buffer will be transmitted. The Ph_Request_Reg $tx\_del_1$ and $tx\_del_0$ bits are used to select what type of delimiter (none, EOF, or EOP) will be appended to the end of the byte transmission. The PEOF delimiter is not available in this state.<br><br>An interrupt is sent to the PL-One firmware as soon as the information in Ph_Request_Reg and Ph_Tx_Buffer has been copied to the SES' internal one-byte transmission buffers. |
| Transition | When the SES has finished transmitting an EOP symbol, it will go to XMIT_CRC. |
| Transition | The SES will stop transmitting and go back to IDLE if the Ph_Request_Reg *tx* bit is cleared. |
| XMIT_CRC State | The PLSE Sublayer is in the process of transmitting 16 CRC symbols. The symbols are generated by the CRC polynomial from the preceding UST values. A CRC bit value of 1 corresponds to the same SUPERIOR state that was used to transmit the PEOF, while a bit value of 0 corresponds to the alternative SUPERIOR symbol. CRC bits are transmitted most significant bit first. For example, if the first Symbol transmitted is SUPERIOR 2 (ie. the PEOF was encoded using SUPERIOR 2s), then a 1 in the CRC buffer = SUPERIOR 2 and 0 = SUPERIOR 1. |
| Action | Once the CRC has been completely transmitted, the SES will send an interrupt to the PL-One firmware. It will also start providing the timestamp to the PL-One firmware, and go to XMIT_STAMP. |
| Transition | The SES will stop transmitting and go back to IDLE if the Ph_Request_Reg *tx* bit is cleared. |
| XMIT_STAMP State | The PLSES is in the process of providing the timestamp to PL-One firmware. The timestamp is the time associated with the end of the CRC transmission. The timestamp will be provided on four bytes (LSB first followed by more significant bytes), where each byte will be sent to PL-One firmware via interrupt in Ph_Rx_Buffer, and the Ph_Confirm_Reg *be/bf* bit will be set. The timestamp is based on a free running counter, which is incremented by the PLSES system clock (i.e. once every 279ns). |
| Transition | The SES will go back to IDLE when Ph_Request_Reg = 0. |

*Table 9 SES State Transition Table*

## Flowcharts for Interfacing to the PLSES

If you wish to program the PL-One yourself, you will find the following flowcharts useful. They document four possible states that the PLSES (and PL-One firmware) can be found in, and show what the expected response is to each of the PLSES Handshake Interrupts.

### High-Level Flowchart

Figure 4 is a high-level flow chart that depicts the operation of the PLSES portion of the PL-One. Four main states exist within the PLSES: Initialization (INIT), Idle (IDLE), Reception (RCV), and Transmission (XMIT). All of these four states are explained in Figure 5 through Figure 8.

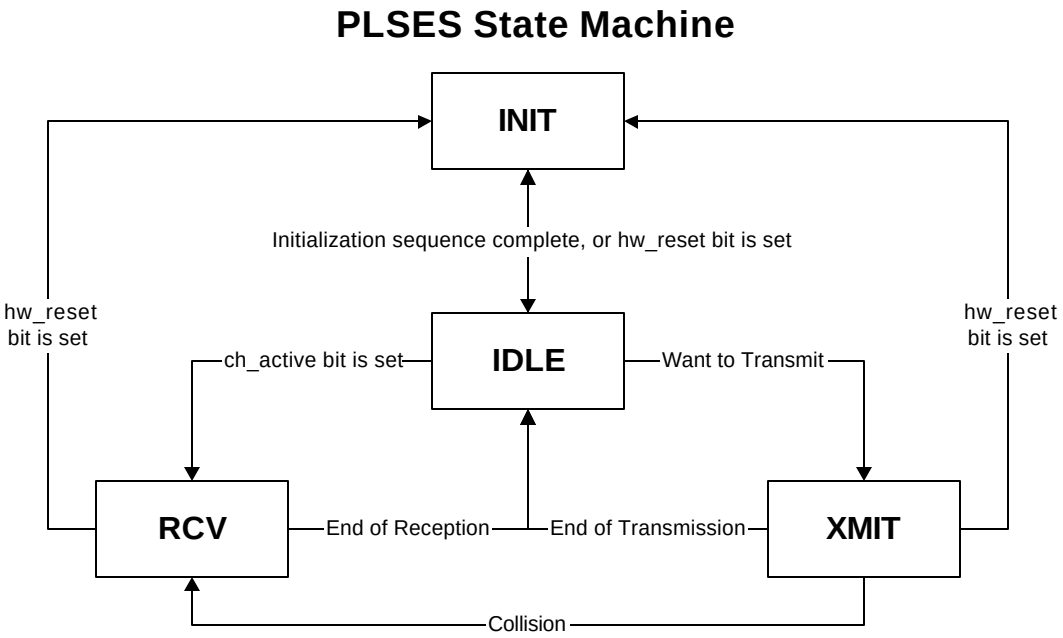



*Figure 4 High-Level Flowchart of PLSES State Machine*

### Initialization (INIT)

This is the state that the CEWay PL-One powers up in, and the state that the PLSES reverts to whenever the *hw_reset* bit is set. No reception or transmission can be done until the initialization sequence is completed, and the PLSES enters the IDLE state.

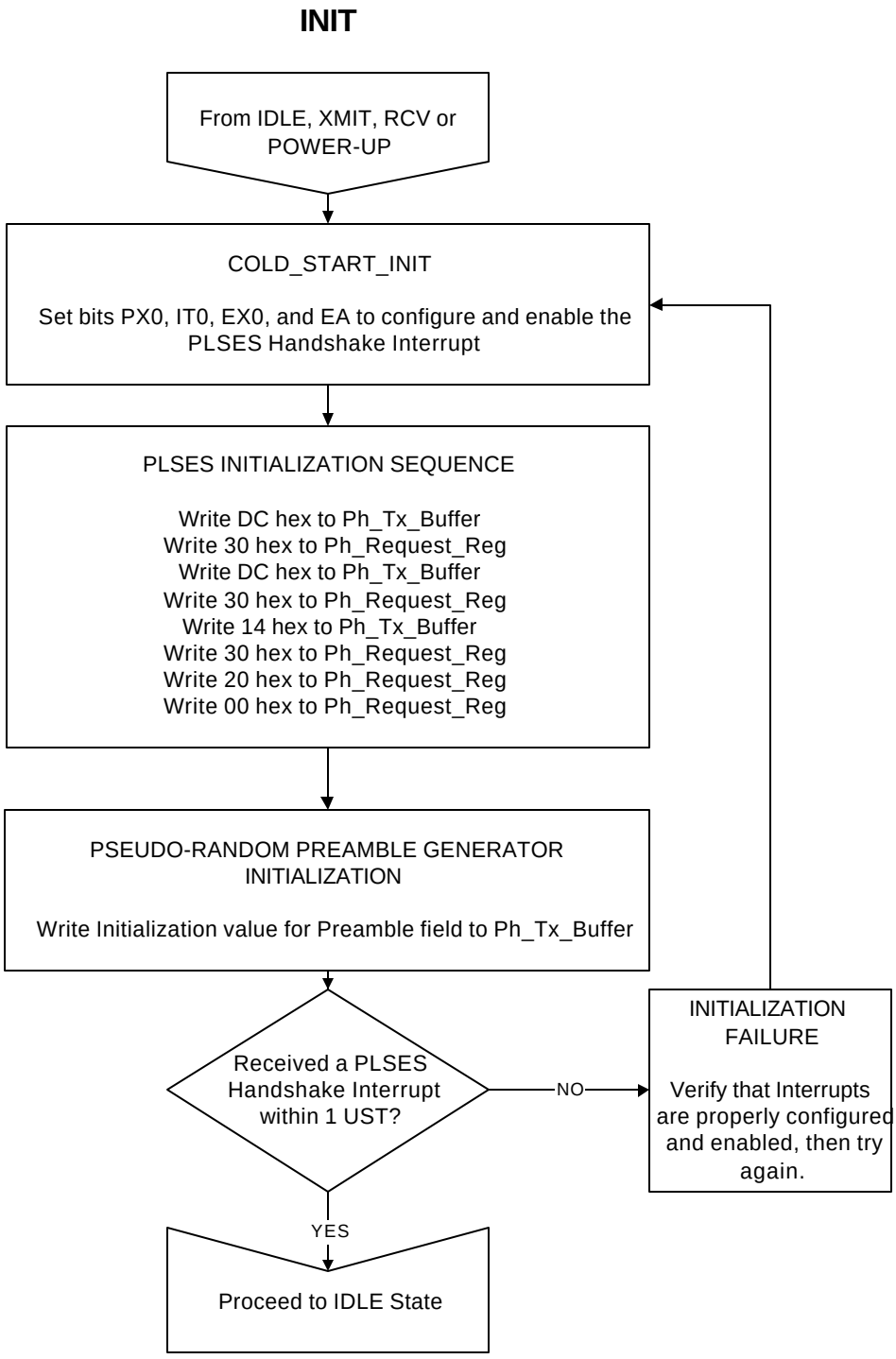



*Figure 5 Intialization Flowchart*

### Idle (IDLE)

This is the default state for the PLSES once it has been initialized. When no transmission or reception is occuring, the PLSES is in the IDLE state. If a PLSES Handshake Interrupt is received with the *ch_active* bit set, then the PLSES has transitioned to the RCV state. Otherwise a transmission request may be made to the PLSES.

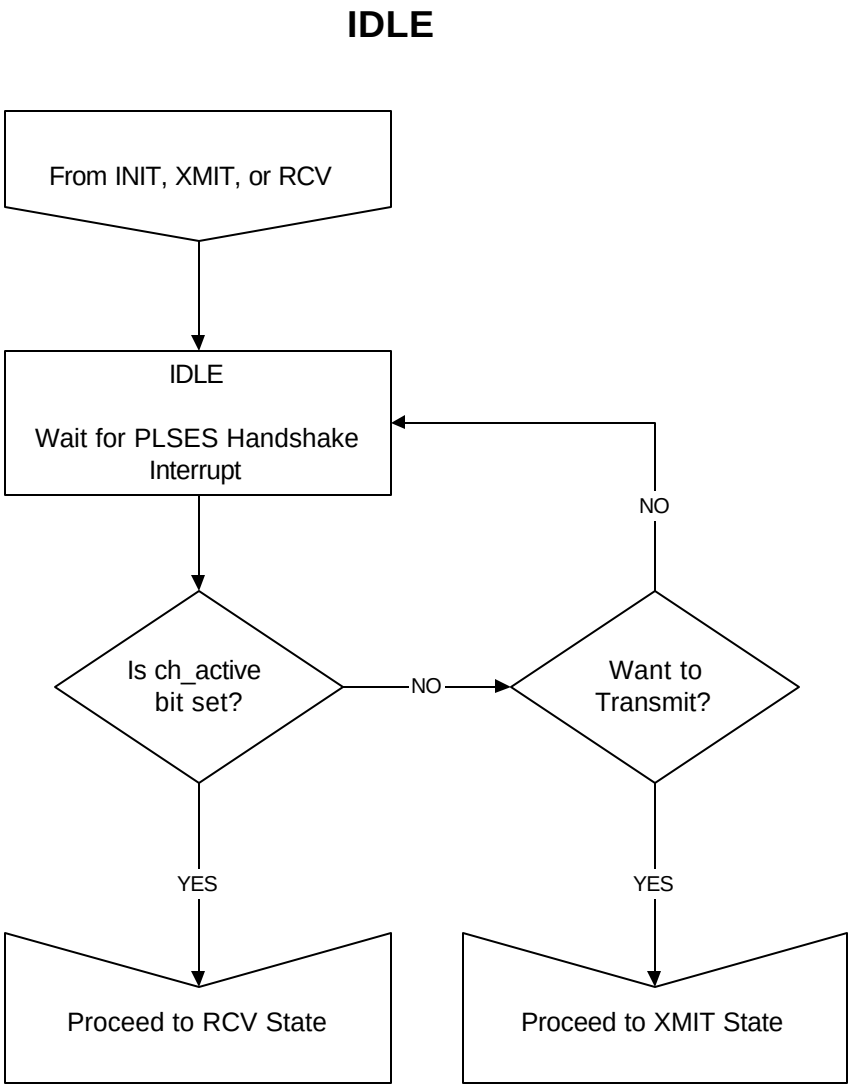



*Figure 6 Idle Flowchart*

### Reception (RCV)

This is the state that the PLSES enters into when it detects a CEBus-like signal on the channel. Explanations of the various actions and decisions found in the flowchart are listed below:

Wait for PLSES Handshake Interrupt: Wait for the PLSES to generate its interrupt (INT0*). Note that in order for the PLSES to consider that the interrupt has been processed and be able to send another interrupt, you need to read the Ph_Confirm_Reg (see PLSES Handshake Interrupt Generation on page 12).

PEOF Received?: If the Ph_Confirm_Reg is 1X1X 0110 bin, this indicates that you have just received the PEOF. If the *ch_noisy* bit is set, it means that the medium was noisy and that only between 3 to 6 PEOF symbols were detected. If it is not set, it means that the medium was not noisy. If the *be/bf* bit (buffer full) is set, it means that the received Preamble symbols are stored in the Ph_Rx_Buffer. If it is not set, it means that there were no received Preamble symbols (probably an acknowledgement packet or an Immediate Retry packet). In this case, the Ph_Rx_Buffer value is irrelevant. If you did not get a PEOF delimiter in the Ph_Confirm_Reg, it means that you are still receiving the Preamble symbols, or have detected a Symbol_Noise condition.

RCV_JABBER_DETECT Detected?: If the *jabber* bit (jabber detected) is set, and the PLSES is in the RCV state, this indicates that the PLSES has received 1000 consecutive SUPERIOR symbols.

Process RCV_JABBER_DETECT: You need to force the PLSES back to the IDLE state by setting the *stop_rx* bit.

SYMBOL_NOISE Detected?: If the *ch_noisy* bit (channel noisy) is set, this means that it was noise on the channel that caused the PLSES to transition to the RCV_PRE_SYM state. Otherwise, you are still receiving the Preamble and/or PEOF.

Process SYMBOL_NOISE: The PLSES automatically transitions from RCV_PRE_SYM to IDLE when a Symbol_Noise condition occurs, and the PL-One firmware should do the same.

CARRIER_LOST Detected?: If the *ch_noisy* bit is set in any PLSES Handshake Interrupt past the reception of the PEOF, it means that a Carrier_Lost condition has occurred. This is when an INFERIOR state is detected on the channel after receiving a PEOF, but before receiving the last CRC symbol.

Process CARRIER_LOST?: A Carrier_Lost condition was detected so the received part of the packet is not complete. Unlike the Symbol_Noise condition, the PLSES does not automatically transition back to the IDLE state when a Carrier_Lost condition occurs. The PL-One firmware needs to update its MACS state variables to deal with the Carrier_Lost condition, and then proceed to the Reset RCV state.

Reset RCV State: Set the *stop_rx* bit (stop reception), which forces the PLSES back to the IDLE state.

Next Byte Received?: If the *be/bf* bit (buffer full) is set, it means that the PLSES received a data byte and it is in the Ph_Rx_Buffer. If it is not set, it means that all that was received was a delimiter. Note that it is possible to receive both a data byte and a delimiter in the same interrupt.

Read Data Byte?: Read one data byte from Ph_Rx_Buffer.

Read Delimiter: Read the delimiter from the Ph_Confirm_Reg_ two *delx* bits (delimiter). The two *delx* bits can be set to 00 bin, which indicates no delimiter, to 01 bin, which indicates an EOF, or to10 bin, which indicates an EOP.

EOP Received?: If your last received data byte was accompanied by the EOP delimiter, then the PLSES will now expect a CRC. If the PLSES has not received the last data byte (EOP), it will simply continue receiving extra data bytes.

Read CRC Status: Read the CRC status from the *gd_packet* bit (good packet). Once the PLSES has finished receiving the CRC it verifies it against an internally generated one. If the CRCs match, then the PLSES sets the *gd_packet* bit.. Otherwise the *gd_packet* bit is not set, indicating that some packet symbols were either missed or misinterpreted, and that the packet should be discarded.

Do you want any RCV_Timestamp Bytes?: You might choose to retrieve the timestamp associated with the reception of your packet's last CRC symbol. It is a 32-bit value with 279 nsec (four or six clock cycles, depending on SELCLK.) It will be passed as four separate bytes in Ph_Rx_Buffer, in the LSByte first order.

Read RCV_Timestamp Byte: Read one timestamp byte from Ph_Rx_Buffer.

Do you want another RCV_Timestamp Byte?: You might choose to retrieve only a partial timestamp, perhaps deducing the values of the most significant bytes. Anywhere between zero and four timestamp bytes may be retrieved.

Request Next RCV_Timestamp Byte: Write 01 hex to Ph_Request_Reg.

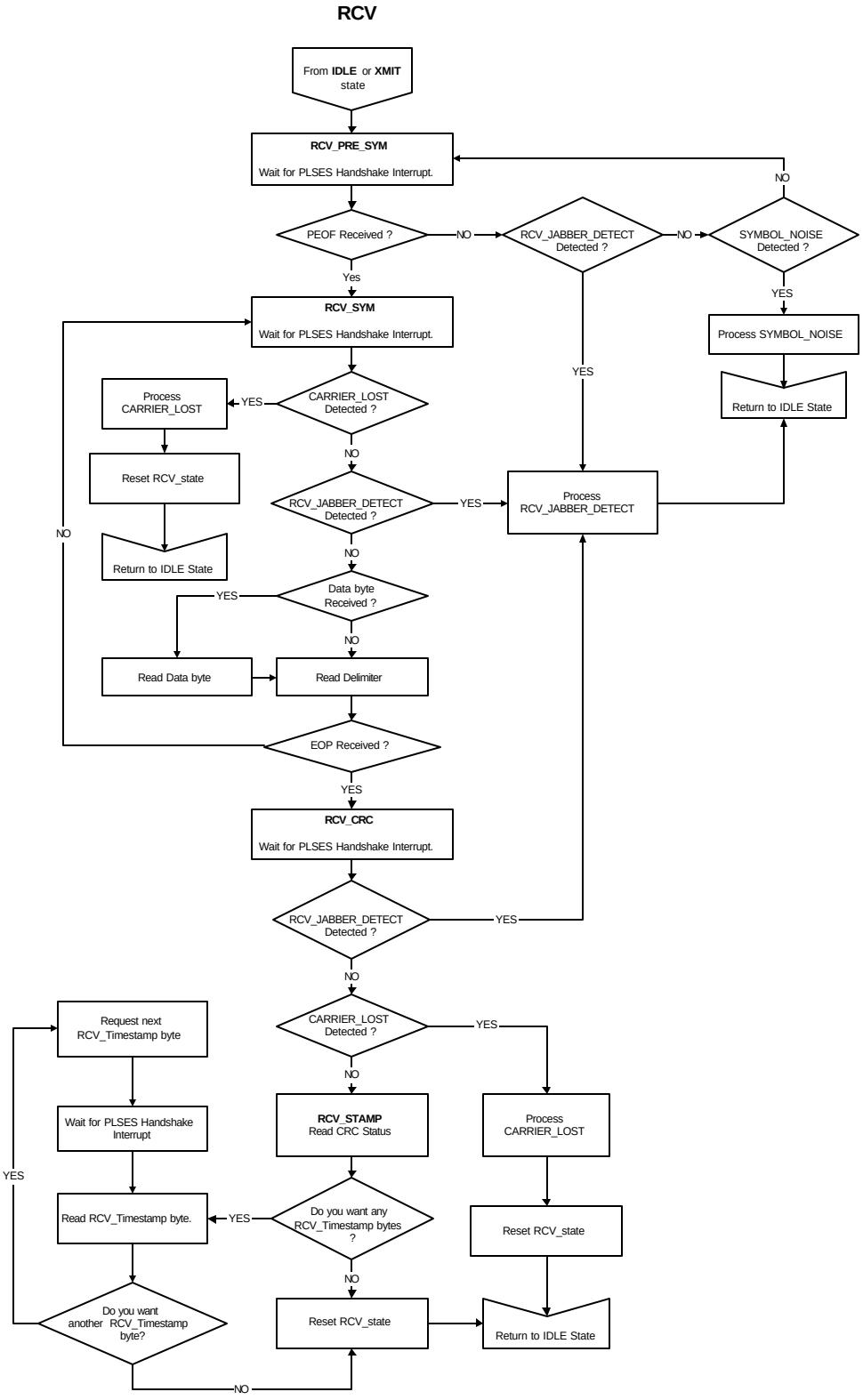



*Figure 7 Reception Flowchart*

### Transmission (XMIT)

This is the state that the PLSES enters into when the PL-One firmware writes the tranmission command to the Ph_Request_Reg. An explanation of the various actions and decisions in the flowchart is listed below:

Transmit Preamble Field: In order to transmit, you need to write 101X 0110 bin to Ph_Request_Reg. If you want to transmit a full Preamble field (8 Preamble symbols and a Preamble End-of-Field (PEOF) delimiter), you need to clear the *lzs* bit (leading zero suppression). If you want to only transmit a PEOF, without a Preamble field, (for acknowledgement packets or for Immediate Retry (IRetry) packets), you need to set the *lzs* bit.

Wait for PLSES Handshake Interrupt: Wait for the PLSES to generate its interrupt (INT0*). Note that in order for the PLSES to consider that the interrupt has been processed and be able to send another interrupt, you need to read the Ph_Confirm_Reg (see PLSES Handshake Interrupt Generation on page 12).

Preamble Field Accepted?: If the Ph_Confirm_Reg *be/bf* bit (buffer empty) is set, it means that the PLSES has stored the pseudo-randomly generated preamble in its buffer and is in the process of transmitting it. If this bit is not set, it means that a collision was detected.

COLLISION Detected?: If the Ph_Confirm_Reg *col/rx* bit (collision) is set, it means that the PLSES has detected a collision while transmitting an INFERIOR Preamble symbol. If this bit is not set, it means that activity was detected on the channel just prior to starting the transmission.

Process PRE-TRANSMISSION COLLISION: An active channel was detected just before the “Transmit Preamble Field” was performed, the PLSES switched into the RCV_PRE_SYM state, and sent an interrupt to the PL-One firmware. For whatever reason, the firmware did not or could not immediately process the interrupt and continued the “Transmit Preamble Field” procedure by writing 101X 0110 bin to Ph_Request_Reg. Since the *stop_rx* bit (10**1**X 0110 bin) was set, this command forces the PLSES back to the IDLE state. The firmware should recognize this transition just like a normal collision.

Process COLLISION: The PLSES has already transitioned from the XMIT to the RCV state and the PL-One firmware should do the same.

Transmit Control Field & EOF: Write the Control field to the Ph_Tx_Buffer and 1011 0010 bin to Ph_Request_Reg. Note that even though you set the *lzs* bit, leading zero suppression will only be performed if there are some leading zeros to suppress. If the top bit of the Control field, the Sequence Number, is set (1), then this byte is transmitted exactly as if the *lzs* bit had not been set.

Control Field Accepted?: If the Ph_Confirm_Reg *be/bf* bit (buffer empty) is set, it means that the PLSES has accepted and stored your Control field in its buffer, is in the process of transmitting it (it has just finished transmitting the PEOF) and is ready for another byte. If the *be/bf* bit is not set, this means that a collision was detected and the *col/rx* bit (collision) will be set.

Transmit Next Byte: Write the next byte to Ph_Tx_Buffer. Then write the appropriate transmission options to Ph_Request_Reg. The base configuration for Ph_Request_Reg is 101X 0XX0 bin. Setting the *lzs* bit will result in the leading zeros in the byte in Ph_Tx_Buffer not being transmitted. The two *delx* bits (delimiter) can be set to 00 bin to select no delimiter, 01 bin to select an EOF or 10 bin to select an EOP. It is not possible to transmit a PEOF delimiter from this state; this delimiter can only be used to begin a transmission.

XMIT_JABBER_DETECT Detected?: If the Ph_Confirm_Reg *jabber* bit (jabber detected) is set and that the PLSES is in the XMIT state, this indicates that the PLSES has transmitted 1000 consecutive SUPERIOR symbols.

Process XMIT_JABBER_DETECT: You must reset the PLSES, via the Ph_Request_Reg *hw_reset* bit (hardware reset). You must then wait for at least 10 000 USTs before attempting another transmission to conform to the CEBus standard.

EOP Requested?: If your last requested "next byte" to transmit was accompanied by the End-of_Packet (EOP) delimiter (last byte to transmit), then, once the PLSES is done transmitting the EOP, it will generate and start transmitting the CRC. If you did not request an EOP delimiter, you need to continue sending other bytes to the PLSES.

Transmit CRC: Write 00 hex in the Ph_Tx_Buffer and 1011 0100 bin in the Ph_Request_Reg. Note that even though this request includes an EOP delimiter, the PLSES will not transmit a second EOP. It will simply generate and transmit the 16-bit CRC.

Do you want any XMIT_Time_Stamp Bytes?: You might choose to retrieve the timestamp associated with the transmission of your packet's last CRC symbol. It is a 32-bit value with 279 nsec (four or six clock cycles, depending on SELCLK.) It will be passed as four separate bytes in Ph_Rx_Buffer, in the LSByte first order.

Read XMIT_Time_Stamp Byte: Read one timestamp byte from Ph_Rx_Buffer.

Do you want another XMIT_Time_Stamp Byte?: You might choose to retrieve only a partial timestamp, perhaps deducing the values of the most significant bytes. Anywhere between zero and four timestamp bytes may be retrieved.

Request Next XMIT_Time_Stamp Byte: Write A1 hex to Ph_Request_Reg.

Reset XMIT State: Write 00 hex in the Ph_Request_Reg.

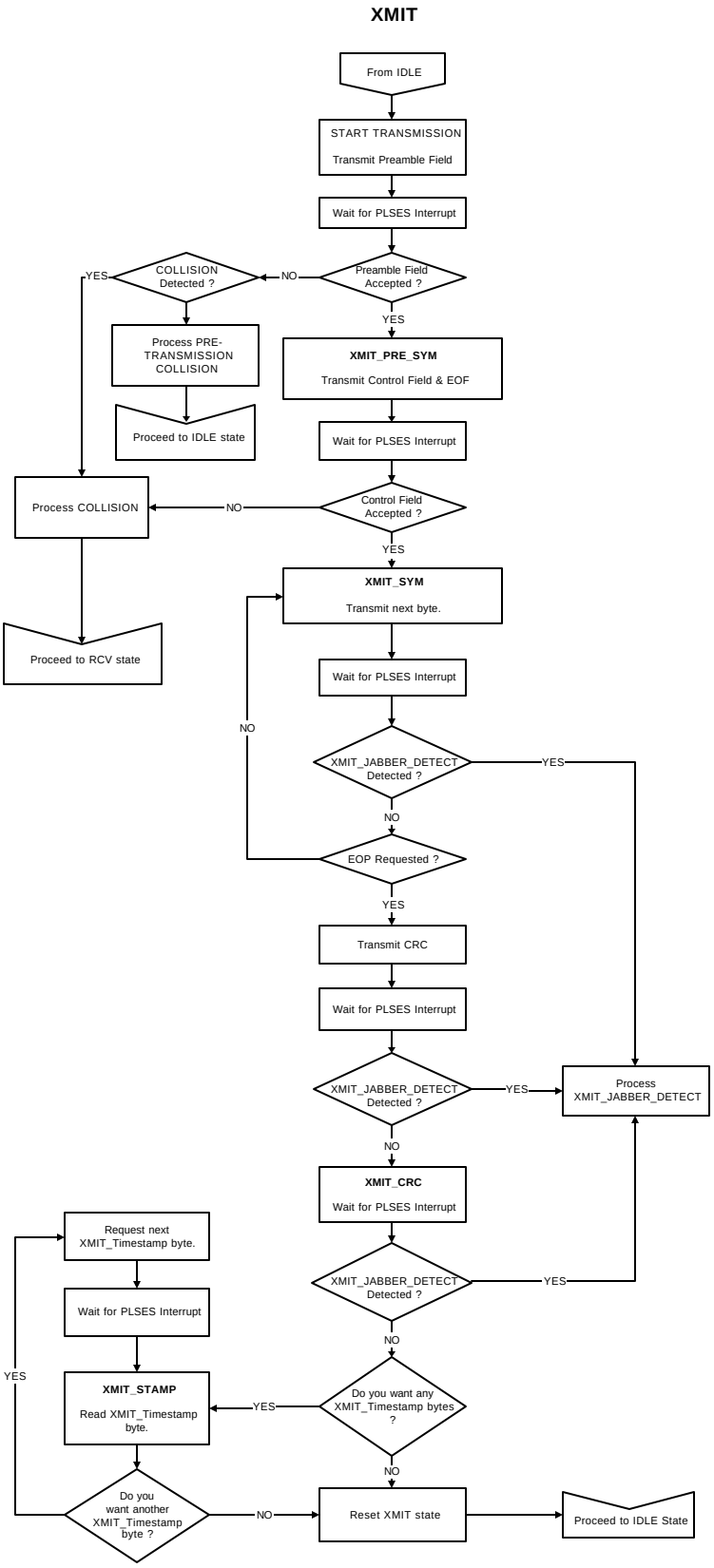



*Figure 8 Transmission Flowchart*

## Timing Constraints

It is important to note that the PLSES sends an interrupt to the PL-One firmware every UST (100 μs) when there isn't any activity on the channel. This interrupt is required if the firmware is to accurately implement the timing constraints specified within the EIA documents on the DLL and MACS (EIA 600.41, 600.42). However it places some severe restrictions on the design of the interrupt service routine. This routine should never last more than 100 μs when in the IDLE state, as an interrupt will then be missed. If possible, it should be considerably shorter than this, so as to allow as much processing time for other processes when the channel is quiet.

### Transmission Timing Constraints

The PLSES uses a one-byte transmission buffer (in addition to Ph_Tx_Buffer) to store the byte that is currently being transmitted. This gives the PL-One firmware some breathing room, as it has between 3 and 26 USTs to write the next byte to the Ph_Tx_Buffer before an underrun condition will occur. For example, when starting transmission, the byte 101X 0000 bin is written to Ph_Request_Reg. As soon as the PLSES starts transmitting the Preamble (or PEOF, if the *lzs* bit was set), it returns a PLSES Handshake Interrupt with the Ph_Confirm_Reg set to 10 hex. The PL-One firmware then has up to the end of the transmission of the PEOF to respond to this Interrupt before an underrun condition will occur. The worst-case scenario in the CEBus standard is when a 00 hex byte is transmitted with leading zero suppression, and an EOF delimiter. Since an EOF takes 3 USTs to transmit, the PL-One firmware has 300 μs to send another byte down to the PLSES before an underrun condition will occur.

### Reception Timing Constraints

The reception timing constraints are similar to the constraints in transmission. The PL-One firmware must ensure that it completes all PLSES Handshake Interrupts within 3 USTs to avoid the possibility of an overrun condition (ex: two EOF symbols immediately after each other, which will occur for any packets sent to the broadcast address.) A much tighter constraint, however, is the CEBus standard requirement that IACK, FAILURE, and ADR_IACK packets be transmitted within 2 USTs of ACK_DATA or ADR_ACK_DATA packets addressed to your local address. It is recommended that the appropriate packet be assembled as soon as it is detected that an acknowledged packet is addressed to you, and that the transmission of this packet should be started in the RCV_STAMP PLSES Handshake Interrupt. This interrupt indicates that the CRC has been received and that the least significant timestamp byte is available in Ph_Rx_Buffer.

## Example

The next page gives an example of an actual packet transmission. The six different values that are visible in Figure 4 are the following:

- CFM_EN: This signal is pulsed every time the PL-One firmware reads the Ph_Confirm_Reg. This is roughly the same as when the PL-One firmware receives a PLSES Handshake Interrupt from the PLSES (there is a small delay between receiving the interrupt and reading the Ph_Confirm_Reg.)
- REQ_EN: This signal is pulsed every time the PL-One firmware writes to the Ph_Request_Reg. This sends an interrupt to the PLSES, which then processes the value written into Ph_Request_Reg.
- PH_CFM: This is the value in the Ph_Confirm_Reg.
- PH_TX: This is the value in the Ph_Tx_Buffer.
- PH_REQ: This is the value in the Ph_Request_Reg.
- M_ST: This is the current medium state requested by the PLSES. This can take one of three values: 0 for INFERIOR, 1 for SUPERIOR Phase 1, and 2 for SUPERIOR Phase 2.

Note: the following explanation will mention several MACS state variables. If you are not familiar with these variables, their definitions can be found in EIA 600.42.

You'll notice that prior to beginning transmission there is a CFM_EN pulse every UST. This is the PLSES Handshake Interrupt indicating that the channel is quiet, and should be used to increment the various MACS transmission counters that are denominated in USTs. Once the Quiet counter is greater or equal to Wait_Time, the PL-One firmware attempts a transmission by writing A6 hex to Ph_Request_Reg. This requests a packet transmission including a full Preamble (since the *lzs* bit was not set.)

The PLSES receives the request, waits until an integral number of USTs have passed since the last time either 00 hex or 20 hex was written to Ph_Request_Reg, and then begins transmission. As soon as transmission begins, the PL-One firmware receives a PLSES Handshake Interrupt indicating that the PLSES is ready for the next byte to transmit (Ph_Confirm_Reg = 10 hex). The PL-One firmware has right up until the end of the transmission of the PEOF to respond to this interrupt by writing the control field to Ph_Tx_Buffer, and B2 hex to Ph_Request_Reg. In this example a value of 0D hex is written to Ph_Tx_Buffer, indicating that this is to be an ADR_ACK_DATA packet.

No collision occurs in this example, as at the end of the tranmission of the PEOF, the MDPS starts transmitting the control field with leading zero suppression (as requested), and the PL-One firmware receives a PLSES handshake interrupt indicating that the transmission buffer is ready for another byte. This process continues through the DA, DHC, SA, SHC, and Data fields, right up until a request is made to transmit the last data byte (09 hex) followed by an EOP symbol (A4 hex to Ph_Request_Reg). Since this transmission information is stored in a one-byte buffer, another PLSES Handshake Interrupt is returned when transmission of this final data byte begins. The PL-One firmware does not need to write to Ph_Tx_Buffer for this interrupt, but does need to write A0 hex to Ph_Request_Reg to keep the transmission going.

Once the EOP has been transmitted, it is followed by a 16-bit CRC. No PLSES Handshake Interrupt is returned for this state transition. When the transmission of the CRC is finished, a final PLSES Handshake Interrupt is returned, after which the PL-One firmware must write 00 hex to Ph_Request_Reg to return the PLSES to its IDLE state. Interrupts will then be received every UST indicating that the channel is quiet, or has become active.

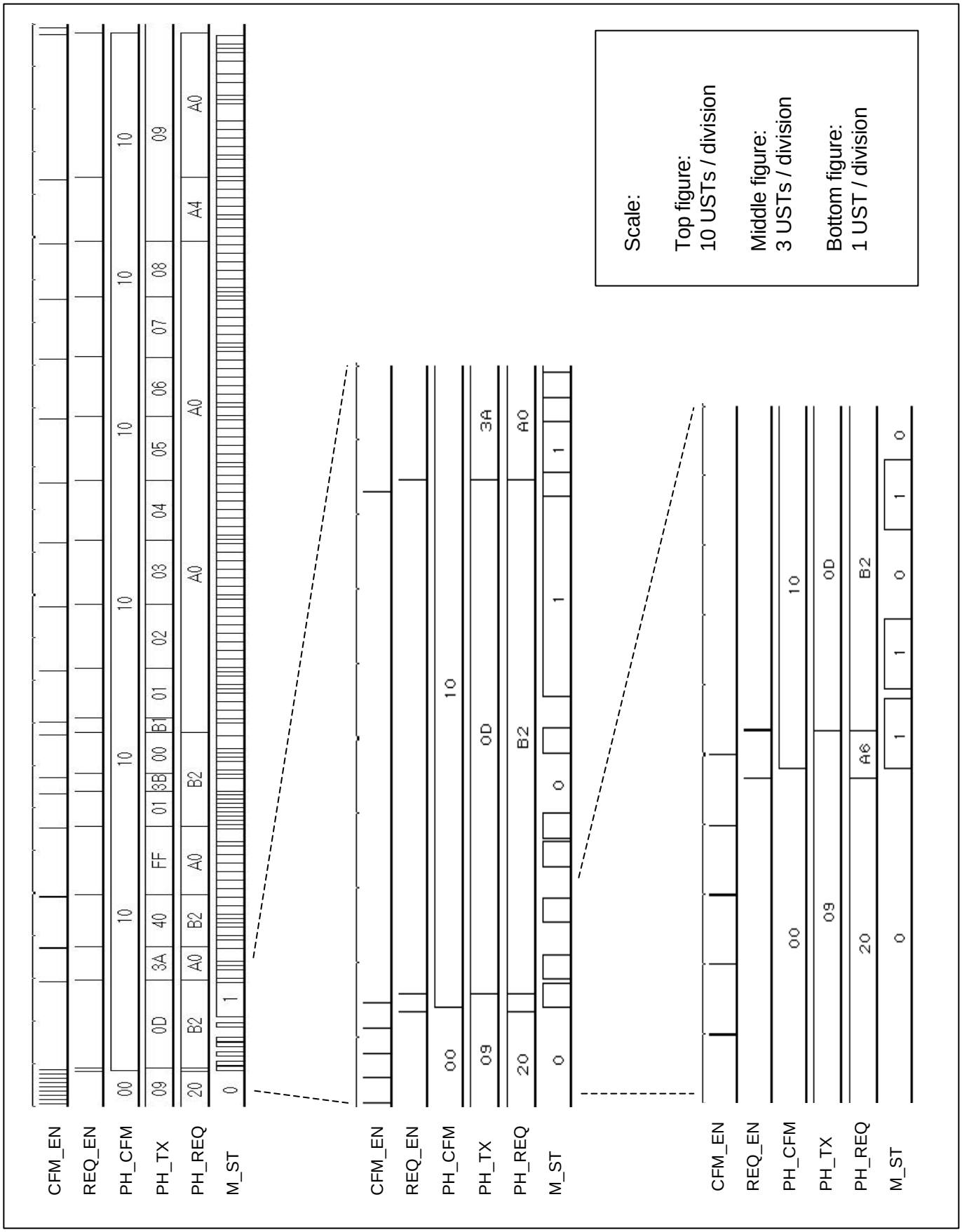



*Figure 9 Example of a Packet Tranmission*

# General Definitions

## Packet Symbols

| | | |
|---|---|---|
| Preamble symbol | ZERO: | 2.28 USTs (SUPERIORs, phase 1, or INFERIOR) |
| Preamble symbol | ONE: | 1.14 USTs (SUPERIORs, phase 1, or INFERIOR) |
| Data symbol | ZERO: | 2 USTs (SUPERIORs, phase 1 or phase 2) |
| Data symbol | ONE: | 1 USTs (SUPERIOR, phase 1 or phase 2) |
| CRC symbol | ZERO: | 1 USTs (SUPERIOR, phase 2) |
| CRC symbol | ONE: | 1 USTs (SUPERIOR, phase 1) |

Packet Delimiters :

| | |
|---|---|
| Preamble EOF (PEOF): | 8 USTs (SUPERIORs, phase 1) |
| End Of Field (EOF): | 3 USTs (SUPERIORs, phase 1 or phase 2) |
| End Of Packet (EOP): | 4 USTs (SUPERIORs, phase 1 or phase 2) |

Ref : EIA 600.31, Section 5.1.1

## Packet Fields

Preamble field, Control field, Destination Address (DA) field, Destination House Code (DHC) field, Source Address (SA) field, Source House Code (SHC) field, Information field and a CRC.

These are the various fields contained in a packet. Note that not all these fields need to be present to form a packet. For example, ADR_IACK packets do not contain the SA or SHC fields; IACK and FAILURE packets do not contain the DA, DHC, SA or SHC fields.

Ref : EIA 600.42, Section 2.2.2, 2.2.4 and 2.2.6.

## Packet Field Parts

Preamble field, Control field, DA low byte, DA high byte, DHC low byte, DHC high byte, SA low byte, SA high byte, SHC low byte, SHC high byte, all Information bytes and the CRC indication.

Note that the term *packet field parts* is used to represent the various parts of a packet, exchanged between the PL-One firmware and PLSES. This is not necessarily a complete packet field since one-byte buffers are used (Ph_Rx Buffer or Ph_Tx Buffer) when exchanging a packet; as well, most fields are longer than one byte in length.

## Symbol Noise

When a noise on the medium has been perceived as a packet symbol. This can be generated either by Preamble symbols having been partly received due to a transmitted partitioned packet, Preamble symbols being received where a sudden noise buries the rest of the packet, or when a real noise is being interpreted as a packet symbol.

# M8052 8-Bit Microcontroller

The M8052 is a high performance 8-bit microcontroller. It is software-compatible (including instruction execution times) with the industry standard 8052AH and 8752BH discrete devices. It can address internal Data Memory of up to 256 bytes. The M8052 can also address 64 kbytes of external Data RAM, and 64 kbytes of external Program ROM via the I/O ports. Three 16-bit timer/counters are provided as well as a full-duplex serial port (UART). The M8052 has 31 external port pins. The M8052 has a power saving mode called Idle mode. In Idle mode, the clock to the CPU is stopped but the timer/counters and serial port are still active.

## MEMORY MAP

### Memory Connection

The M8052 has separate program and data memory. Both internal and external data memory can be accessed, while only external program memory can be accessed. External memory accesses use port 0 as a multiplexed address and data bus and port 2 for the high order address lines. ALE is used to latch the lower-order address which appears on port 0. PSEN* is the program memory read strobe, RD* is the data memory read strobe and WR* is the data memory write strobe.

### Program Memory

Up to 64 kbytes of external program memory can be accessed directly. More can be accessed if a bank-switching scheme is implemented.

### Data Memory

#### *Internal Data Memory*

Internal data memory for the M8052 is 256 bytes. The whole internal data memory is accessible using indirect addressing, but only the lower 128 bytes are accessible using direct addressing. The upper 128 bytes of direct address data memory space are used to access SFRs (see below).

#### *Register Banks*

There are four directly addressable register banks switched between using the PSW (see below), which occupy data memory space from 00 hex to 1F hex.

### *Bit Addressing*

16 bytes of data memory (addresses 20 hex to 2F hex) are bit addressable. SFRs that have addresses of 1XXX X000 bin are bit addressable.

### *Scratch Pad*

Direct addresses from 3F hex to 7F hex are usable as scratch pad registers or for a stack.

### *External Data Memory*

External data memory may be either 8-bit or 16-bit addressable. Registers R0 and R1 are used for indirect 8-bit addressing, the DPTR register is used for 16-bit addressing.

## SPECIAL FUNCTION REGISTERS

All I/O, timer/counter and UART operations for the M8052 are accessed via Special Function Registers (SFRs). These registers occupy direct data memory space locations in the range 80 hex to FF hex. In the same area various control and program registers may be accessed. Their names and addresses are given in Table 10 below, and their definitions follow.

**SFR Map**

| Description | Abbreviation | Address | Bit Addressable |
|---|---|---|---|
| Port 0 | P0 | 80 | ✓ |
| Stack Pointer | SP | 81 | |
| Data Pointer Low Byte | DPL | 82 | |
| Data Pointer High Byte | DPH | 83 | |
| Receive Buffer from PLSES | PH_RX_BUF | 84* | |
| Transmit Buffer to PLSES | PH_TX_BUF | 85* | |
| Power Control Register | PCON | 87 | |
| Timer/Counter Control | TCON | 88 | ✓ |
| Timer/Counter Mode Control | TMOD | 89 | |
| Timer/Counter 0 low byte | TL0 | 8A | |
| Timer/Counter 1 low byte | TL1 | 8B | |
| Timer/Counter 0 high byte | TH0 | 8C | |
| Timer/Counter 1 high byte | TH1 | 8D | |
| Port 1 | P1 | 90 | ✓ |
| Serial Control Register | SCON | 98 | ✓ |
| Serial Data Buffer | SBUF | 99 | |
| Port 2 | P2 | A0 | ✓ |
| Interrupt Enable Register | IE | A8 | ✓ |
| Port 3 | P3 | B0 | ✓ |
| Interrupt Priority Register | IP | B8 | ✓ |
| Timer/Counter 2 Control | T2CON | C8 | ✓ |
| Capture Register 2 low byte | RCAP2L | CA | |
| Capture Register 2 high byte | RCAP2H | CB | |

| | | | |
|---|---|---|---|
| Timer/Counter 2 low byte | TL2 | CC | |
| Timer/Counter 2 high byte | TH2 | CD | |
| Program Status Word | PSW | D0 | ✓ |
| Confirm Port from PLSES | PH_CONFIRM_ REG | D8* | ✓ |
| Accumulator | A | E0 | ✓ |
| Request Port to PLSES | PH_REQUEST_ REG | E8* | ✓ |
| B Register | B | F0 | ✓ |

*Table 10 M8052 SFR Map*

Note that the registers marked * are not present in the original device.

**SFR Definitions**

***I/O Ports (P0 P1 P2 P3)***

P0, P1, P2, P3.0 and P3.1 are latches used to drive the 26 quasi-bidirectional I/O lines. On reset they are all set to the FF hex value, which is input mode. P3.3 and P3.5 are always in input mode; writing these bits has no effect on the external port or on the next value read from these bits. P3.4, P3.6 and P3.7 are always in output mode. External voltages applied to these pins can not be read from within the chip.

***Stack Pointer (SP)***

The SP register contains the stack pointer. The stack pointer is used to load the program counter into memory during LCALL and ACALL instructions and is used to retrieve the program counter from memory in RET and RETI instructions. Temporary data may also be saved or loaded using PUSH and POP instructions, which also increment and decrement the stack pointer. The stack pointer points to the top location of the stack. On reset the stack pointer is set to 07 hex.

***Data Pointer (DPTR)***

The Data Pointer (DPTR) is 16 bits in size and consists of two registers, the Data Pointer High byte (DPH), and the Data Pointer Low byte (DPL). Two 16-bit operations are possible on this register, they are load immediate and increment. This register is used for 16-bit address external memory accesses, for offset code byte fetches and for offset program jumps. On reset, the value of this register is 0000 hex.

***Power Control Register (PCON)***

Bit definitions for this register are as follows:

**PCON.7** SMOD. Double baud rate bit. For use see below.
**PCON.6** Not implemented.
**PCON.5** Not implemented.
**PCON.4** Not implemented.
**PCON.3** GF1. General purpose flag bit.
**PCON.2** GF0. General purpose flag bit.
**PCON.1** Not implemented.
**PCON.0** IDL. Idle bit. If 1, Idle mode is entered.

On reset this register returns 0XXX0000 binary.

### *Timer/Counter Registers*

Three 16-bit timer counters are provided. TCON and TMOD are used to set the mode of operation and to control the running and interrupt generation of timer counters 0 and 1. T2CON controls operation of timer counter 2. The timer/counter values are stored in three pairs of 8-bit registers (TL0, TH0, TL1, TH1, and TL2, TH2). The value stored in timer counter 2 may be "captured" into RCAP2L and RCAP2H.

### *Timer/Counter Control (TCON)*

Bit definitions for this register are as follows:

**TCON.7** TF1. Timer 1 overflow flag. Set by hardware when timer/counter 1 overflows. Cleared by hardware when processor calls the interrupt service routine.

**TCON.6** TR1. Timer 1 run control. If 1, timer runs. If 0, timer is halted.

**TCON.5** TF0. Timer 0 overflow flag. Set by hardware when timer/counter 0 overflows. Cleared by hardware when processor calls the interrupt service routine.

**TCON.4** TR0. Timer 0 run control. If 1, timer runs. If 0, timer is halted.

**TCON.3** IE1. External interrupt 1 edge flag. Set by hardware when an External Interrupt 1 edge is detected.

**TCON.2** IT1. Interrupt 1 control bit. If 1, a falling edge triggers an interrupt. If 0, a low level triggers an interrupt.

**TCON.1** IE0. External interrupt 0 edge flag. Set by hardware when an External Interrupt 0 edge is detected.

**TCON.0** IT0. Interrupt 0 control bit. If 1, a falling edge triggers an interrupt. If 0, a low level triggers an interrupt. This bit ***must*** be set to 1 for correct operation of the PLSES interrupt.

### *Timer/Counter Mode (TMOD)*

**TMOD.7** GATE1. Timer 1 gate flag. When TCON.6 is set and GATE1 = 1, timer/counter 1 will only run if INT1 pin is 1 (hardware control). When GATE1 = 0, timer/counter 1 will only run if TCON.6 = 1 (software control).

**TMOD.6** C/NT1. Timer/Counter 1 selector. If 0 input is from internal system clock, if 1 input is from T1 pin.

**TMOD.5** M1(1). Timer 1 Mode control bit M1.

**TMOD.4** M0(1). Timer 1 Mode control bit M0.

**TMOD.3** GATE0. Timer 0 gate flag. When TCON.6 is set and GATE0 = 1, timer/counter 1 will only run if INT0 pin is 1 (hardware control). Note that this option is not very useful, since the value of the INT0 pin

(P3.2) is controlled by the PLSES. When GATE0 = 0, timer/counter 0 will only run if TCON.6 = 1 (software control).

**TMOD.2** C/NT0. Timer/Counter 0 selector. If 0, input is from internal system clock. If 1, input is from T0 pin. This bit ***must*** be set to 0 for correct operation of Timer 0, as pin T0 is output only.

**TMOD.1** M1(0). Timer 0 Mode control bit M1.

**TMOD.0** M0(0). Timer 0 Mode control bit M0.

For both timer counters the mode bits M0 and M1 apply as follows:

| M1 | M0 | Operating Mode |
|---|---|---|
| 0 | 0 | 13-bit timer/counter (M8048 compatible mode). |
| 0 | 1 | 16-bit timer/counter. |
| 1 | 0 | 8-bit auto-reload timer/counter. |
| 1 | 1 | Timer 0 is split into two halves. TL0 is an 8-bit timer/counter controlled by the standard timer 0 control bits. TH0 is an 8-bit timer/counter controlled by the standard timer 1 control bits. TH1 and TL1 are held (Timer 1 is stopped). |

*Table 11 Configuration Mode Bits (TMOD.5, TMOD.4, TMOD.1, TMOD.0) for Timers 0 and 1*

**Timer/Counter Data (TL0 TL1 TH0 TH1)**
TL0 and TH0 are the low and high bytes of timer/counter 0 respectively. TH0 and TH1 are the low and high bytes of timer counter 1, respectively. On reset all timer/counter registers are 00 hex.

***Timer/Counter 2 Control (T2CON)***
Bit definitions for this register are as follows:

**T2CON.7** TF2. Timer 2 overflow flag. Set by hardware when timer/counter 2 overflows unless either RCLK or TCLK is set to 1. This bit is not cleared by hardware when the processor calls the interrupt service routine.

**T2CON.6** EXF2. Timer 2 external flag. This bit is set when a capture or reload is triggered by a negative transition on T2EX (P1.1), and EXEN2 is set to 1. If Timer 2 interrupt is enabled, setting this bit will cause an interrupt to the Timer 2 vector.

**T2CON.5** RCLK. If this bit is set, the Serial Port receive clock is driven from the overflow pulses of Timer 2.

**T2CON.4** TCLK. If this bit is set, the Serial Port transmit clock is driven from the overflow pulses of Timer 2.

**T2CON.3** EXEN2. Timer 2 External interrupt enable flag. When set, a negative edge on T2EX (P1.1) triggers a capture or auto-reload.

**T2CON.2** TR2. Run control bit for Timer 2. If set to 1, the timer is enabled.

**T2CON.1** C/NT2. Timer/Counter select. A 0 selects internal timer mode. A 1 selects external counter mode.

**T2CON.0** CP/NRL2. Capture/Reload control. When set, captures occur on negative transitions of T2EX (if EXEN2 is set). If 0, auto-reloads are performed on timer overflows or on negative transitions of T2EX (if EXEN2 is set). If either RCLK or TCLK is 1, this bit is ignored and auto-reloads are performed on timer overflows.

**Timer/Counter 2 Data (TL2 TH2 RCAP2L RCAP2H)**

TL2 and TH2 are the low and high bytes of timer/counter 2 respectively. RCAP2L and RCAP2H are the low and high bytes of timer 2 capture registers. These registers are also used for auto-reload. TH2 and TL2 cannot be reliably read from or written to while Counter/Timer 2 is operating as a timer in either baud-rate mode. Timer 2 should be turned off using T2CON before reading or writing to the counter registers in this mode. On reset all these registers return 00 hex.

**UART Registers**

The UART uses two SFRs. SCON is the control register, SBUF the data register. Data is written to SBUF for transmission, and SBUF is read to obtain received data. The received data and transmitted data registers are completely independent. The bit definition for SCON follows.

***UART Control (SCON)***

**SCON.7** SM0. UART mode specifier.

**SCON.6** SM1. UART mode specifier.

**SCON.5** SM2. UART mode specifier.

**SCON.4** REN. If 1, enables reception. If 0, disables reception.

**SCON.3** TB8. In modes 2 and 3, this is the 9th data bit sent.

**SCON.2** RB8. In modes 2 and 3, this is the 9th data bit received. In mode 1, if SM2 = 0, this is the stop bit received. In mode 0, this bit is not used.

**SCON.1** TI. Transmit interrupt flag. This is set by hardware at the end of the 8th bit in mode 0, or at the beginning of the stop bit in other modes. Must be cleared by software.

**SCON.0** RI. Receive interrupt flag. This is set by hardware at the end of the 8th bit in mode 0, or at the half point of the stop bit in other modes. Must be cleared by software. The mode control bits operate as follows:

| SM0 | SM1 | Operating Mode | Baud Rate |
|---|---|---|---|
| 0 | 0 | Mode 0: 8-bit shift register | $f_{osc}$ /12 |
| 0 | 1 | Mode 1: 8-bit UART | Variable |
| 1 | 0 | Mode 2: 9-bit UART | $f_{osc}$ /32 or $f_{osc}$ /64 |
| 1 | 1 | Mode 3: 9-bit UART | Variable |

*Table 12 UART Mode Bits (SCON.7, SCON.6) for Serial Port*

SM2 modifies the above as follows: In modes 2 & 3, if SM2 is set, the receive interrupt will not be generated if the received 9th data bit is 0. In mode 1, the receive interrupt will not be generated if a valid stop bit is not received. In mode 0, SM2 should be 0. This bit enables multi-processor communication over a single serial line.

### *UART Baud Rates*

The UART baud rates in each of the 4 modes are defined as follows:

**Mode 0.** The baud rate is fixed:

Baud Rate = $f_{osc}$ / 12.

**Modes 1 and 3.** The baud rate is variable, determined by timer/counter 1 or timer/counter 2 (timer/counter 1 should be in auto-reload mode, timer/counter 2 should be in baud rate generating mode):

Baud Rate = K × $f_{osc}$ / 32 × 12 × (256 - TH1)

**Mode 2.** The baud rate is selectable from one of two values: Baud Rate = K × $f_{osc}$ / 64

K = PCON.7 + 1.
TH1 is the auto-reload value for timer/counter 1.
It should be noted that in modes 1 and 3 the equation is valid for all values of TH1.
If Timer 2 is being clocked internally, then:
Timer 2 Baud Rate = fosc / 32 × ( 65536 - (RCAP2H.RCAP2L) )
The above equation is valid for all values of RCAP2 except 65535 (FFFFh).
If Timer 2 is being clocked internally, then:
Timer 2 Baud Rate = Timer 2 Overflow Rate / 16
The above equation is valid for all values of RCAP2.

### *UART Data (SBUF)*

This register is used for both transmit and receive data. Transmit data is written to this location and receive data is read from this location, but the two paths are independent. On reset SCON returns 00 hex, SBUF returns XX hex.

### *Interrupt Enable Register (IE)*

For each bit in this register, a 1 enables the corresponding interrupt, and a 0 disables it. The allocation of interrupts to bits is as follows:

**IE.7** EA. Enable or disable all interrupt bits
**IE.6** Not implemented
**IE.5** ET2. Enable or disable Timer 2 interrupt
**IE.4** ES. Enable or disable serial port interrupt
**IE.3** ET1. Enable or disable Timer 1 overflow interrupt
**IE.2** EX1. Enable or disable External interrupt 1
**IE.1** ET0. Enable or disable Timer 0 overflow interrupt
**IE.0** Ex0. Enable or disable PLSES interrupt
On reset, this register returns 00000000 binary.

### *Interrupt Priority Register (IP)*

For each bit in this register, a 1 selects high priority for the corresponding interrupt, and a 0 selects low priority. The allocation of interrupts to bits is as follows:

**IP.7** Not implemented
**IP.6** Not implemented
**IP.5** PT2. Select priority for Timer 2 interrupt
**IP.4** PS. Select priority for serial port interrupt
**IP.3** PT1. Select priority for Timer 1 overflow interrupt
**IP.2** PX1. Select priority for External interrupt 1
**IP.1** PT0. Select priority for Timer 0 overflow interrupt
**IP.0** Px0. Select priority for PLSES interrupt. This should always be set to 1 to ensure correct data transfer between the 8052 and the PLSES.

When an interrupt is in progress, only a higher level interrupt can interrupt the service routine. On reset, this register returns 00000000 binary.

### *Program Status Word (PSW)*

This register contains status information resulting from CPU and ALU operation. The bit definitions are given below:

**PSW.7** CY. ALU carry flag
**PSW.6** AC. ALU auxiliary carry flag
**PSW.5** F0. General purpose user definable flag
**PSW.4** RS1. Register bank select bit 1
**PSW.3** RS0. Register bank select bit 0
**PSW.2** OV. ALU overflow flag
**PSW.1** F1. User definable flag
**PSW.0** P. Parity flag. Set each instruction cycle to indicate odd/even parity in the accumulator**.**

The register bank select bits operate as follows.

| RS1 | RS0 | Register Bank Select |
|---|---|---|
| 0 | 0 | RB0. Registers from 00 - 07 hex. |
| 0 | 1 | RB1. Registers from 08 - 0F hex. |
| 1 | 0 | RB2. Registers from 10 - 17 hex. |
| 1 | 1 | RB3. Registers from 18 - 1F hex. |

*Table 13 Register Bank Selection Bits (PSW.4, PSW.3)*

On reset, this register returns 00 hex.

### *Accumulator (ACC)*

This register provides one of the operands for most ALU operations. In the instruction table below it is denoted as "A".

On reset, this register returns 00 hex.

### *B Register (B)*

This register provides the second operand for multiply or divide instructions. Otherwise it may be used as a scratch pad register.

On reset, this register returns 00 hex.

## INTERRUPTS

The M8052 provides 6 interrupt sources. The external interrupt INT1* is either level or edge triggered (depending on bits in TCON, see above). Timer 0, 1 and 2 interrupts are generated by TF0, TF1 and TF2 via a rollover in their respective registers (or a negative transition on T2EX), except in mode 3 when TH0 controls timer 1 interrupt. The serial interrupt is generated by a logical OR of RI and TI. The Timer 2 interrupt is generated by the logical OR of TF2 and EXF2.

The CEWay PL-One uses the standard INT0* by the PLSES block, so this interrupt will not be accessible via an external pin. Details on when this interrupt is generated are described in the PLSES section. All other characteristics of the interrupt services are described in this section.

### Interrupt Flag Clear

If the external interrupts are edge triggered, the interrupt flag is cleared on vectoring to the service routine, but if they are level triggered then the flag is controlled by the external signal. Timer counter flags 0 and 1 are cleared on vectoring to the interrupt service routine, but the serial interrupt flag is not affected by hardware. The serial interrupt flag and Timer 2 flag are not affected by hardware, they should be cleared by software.

### Priority Levels

One of two priority levels may be selected for each interrupt. A high priority interrupt may interrupt the service routine of a low priority interrupt, and if two interrupts of different priority occur at the same time the higher level interrupt will be serviced first. An interrupt cannot be interrupted by another interrupt of the same priority level. If two interrupts of the same priority level occur simultaneously, a polling sequence is observed as follows:

| Source | Level | Description |
|---|---|---|
| IE0 | 1 (highest) | PLSES Interrupt |
| TF0 | 2 | Timer/Counter Interrupt 0 |
| IE1 | 3 | External Interrupt 1 |
| TF1 | 4 | Timer/Counter Interrupt 1 |
| RI + TI | 5 | Serial Interrupt |
| TF2 + EXF2 | 6 (lowest) | Timer/Counter 2 Interrupts |

*Table 14 Interrupt Priority Levels and Descriptions*

**Interrupt Vectors**

When an interrupt is serviced, a long call instruction is executed to one of the following locations:

| Source | Vector Address |
|---|---|
| IE0 | 0003 Hex |
| TF0 | 000B Hex |
| IE1 | 0013 Hex |
| TF1 | 001B Hex |
| RI + TI | 0023 Hex |
| TF2 + EXF2 | 002B Hex |

*Table 15 Interrupt Vector Addresses*

**Interrupt Latency**

The response time in a single interrupt system is between 3 and 9 machine cycles, depending on which instruction was being executed when the interrupt occurred. This can be reduced to 7 machine cycles if the MUL or DIV instructions are never used.

# INSTRUCTION DEFINITIONS

The M8052 instruction set is shown in Table 17 and some of the features supported are outlined below.

### Addressing Modes

The set provides a variety of addressing modes which are outlined below.

### Direct Addressing

In direct addressing, the operand is specified by an 8-bit address field. Only internal data and SFRs may be accessed using this mode.

### Indirect Addressing

In indirect addressing, the operand is specified by an address contained in a register. Two registers (R0 and R1) from the current bank, or the DPTR may be used for addressing in this mode. Both internal and external data memory may be indirectly addressed.

### Register Addressing

In register addressing, the operand is specified by the top 3 bits of the opcode, which selects one of the current bank of registers. Four banks of registers are available. The current bank is selected by bits in the PSW.

### Register Specific Addressing

Some instructions only operate on specific registers, and this may be defined in the opcode. In particular, some accumulator operations and some pointer operations are defined in this manner.

### Immediate Data

Instructions which use immediate data are 2 bytes long or over, and the immediate operand is stored in program memory as part of the instruction.

### Indexed Addressing

Only program memory may be addressed using indexed addressing. It is intended for simple implementation of lookup tables. A 16-bit base register (either the PC or the DPTR) is combined with an offset stored in the accumulator to access data in program memory.

### Arithmetic Instructions

The M8052 implements ADD, ADDC (Add with Carry), SUBB (Subtract with Borrow), INC (Increment) and DEC (Decrement) functions, which may be used in most addressing modes. There are three accumulator specific instructions, DA A (Decimal Adjust A), MUL AB (Multiply A by B) and DIV AB (Divide A by B).

### Logical Instructions

The M8052 implements ANL (AND Logical), ORL (OR Logical) and XRL (Exclusive-OR Logical) functions, which again may be used in most addressing modes. There are seven accumulator specific instructions, CLR A (Clear A), CPL A (Complement A), RL A (Rotate Left A), RLC A (Rotate Left through Carry A), RR A (Rotate Right A), RRC A (Rotate Right through Carry A) and SWAP A (Swap Nibbles of A).

## Data Transfers

### Internal Memory

Data may be moved from the accumulator to any internal memory location, from any internal memory location to the accumulator, and from any internal memory location to any SFR or other internal memory location.

### External Memory

The accumulator data may be moved to or from an external memory location in one of two addressing modes. In 8-bit addressing mode, the external location is addressed by either R0 or R1; in 16-bit addressing mode the location is addressed by the DPTR.

## Jump Instructions

### Unconditional Jumps

Four kinds of unconditional jump instructions are available. Short jumps (SJMP) are relative jumps (limited to -128 to +127 bytes). Long jumps (LJMP) are absolute 16-bit jumps, and Absolute jumps (AJMP) which are absolute 13-bit jumps (i.e. within a 2k byte memory page). The last type is an indexed jump, JMP @A+DPTR, which jumps to a location contained in the DPTR register, offset by value stored in the accumulator.

### Subroutine Calls and Returns

There are only two sorts of subroutine call, ACALL and LCALL, which are Absolute and Long as above. Two return instructions are provided, RET and RETI, the latter being for interrupt service routines.

### Conditional Jumps

Conditional jump instructions all use relative addressing, so are limited to the same -128 to +127 byte range as above.

### Boolean Instructions

The bit addressable registers in both direct and SFR space may be manipulated using boolean instructions. There are available logical functions which use the carry flag and an addressable bit as the operands, and each addressable bit may be set, cleared or tested in a jump instruction.

### Flags

Instructions that affect flags generated by the ALU are as follows.

| Instruction | Flag | | |
|---|---|---|---|
| | C | OV | AC |
| ADD | X | X | X |
| ADDC | X | X | X |
| SUBB | X | X | X |
| MUL | 0 | X | |
| DIV | 0 | X | |
| DA | X | | |
| RRC | X | | |

| | | | |
|---|---|---|---|
| RLC | X | | |
| SETB C | 1 | | |
| CLRC | 0 | | |
| CPLC | X | | |
| ANL C, bit | X | | |
| ANL C, /bit | X | | |
| ORL C, bit | X | | |
| ORL C, /bit | X | | |
| MOV C, bit | X | | |
| CJNE | X | | |

*Table 16 Instructions that Affect ALU Flag Settings*

In the above table, a 0 means the flag is always cleared, a 1 means the flag is always set, and an X means that the state of the flag depends on the result of the operation.

## Instruction Table

| **ARITHMETIC** | | | | |
|---|---|---|---|---|
| **Mnemonic** | **Description** | **Bytes** | **Cycles** | **Hex Code** |
| ADD A,Rn | Add register to A | 1 | 1 | 28-2F |
| ADD A,dir | Add direct byte to A | 2 | 1 | 25 |
| ADD A,@Ri | Add data memory to A | 1 | 1 | 26-27 |
| ADD A,#data | Add immediate to A | 2 | 1 | 24 |
| ADDC A,Rn | Add register to A with carry | 1 | 1 | 38-3F |
| ADDC A,dir | Add direct byte to A with carry | 2 | 1 | 35 |
| ADDC A,@Ri | Add data memory to A with carry | 1 | 1 | 36-37 |
| ADDC A,#data | Add immediate to A with carry | 2 | 1 | 34 |
| SUBB A,Rn | Subtract reg. from A with borrow | 1 | 1 | 98-9F |
| SUBB A,dir | Subtract direct from A with borrow | 2 | 1 | 95 |
| SUBB A,@Ri | Subtract indirect from A with borrow | 1 | 1 | 96-97 |
| SUBB A,#data | Subtract imm. from A with borrow | 2 | 1 | 94 |
| INC A | Increment A | 1 | 1 | 04 |
| INC Rn | Increment register | 1 | 1 | 08-0F |
| INC dir | Increment direct byte | 2 | 1 | 05 |
| INC @Ri | Increment data memory | 1 | 1 | 06-07 |
| DEC A | Decrement A | 1 | 1 | 14 |
| DEC Rn | Decrement register | 1 | 1 | 18-1F |
| DEC dir | Decrement direct byte | 2 | 1 | 15 |
| DEC @Ri | Decrement data memory | 1 | 1 | 16-17 |
| INC DPTR | Increment data pointer | 1 | 2 | A3 |
| MUL AB | Multiply A by B | 1 | 4 | A4 |
| DIV AB | Divide A by B | 1 | 4 | 84 |
| DA A | Decimal Adjust A | 1 | 1 | D4 |
| **LOGICAL** | | | | |
| ANL A,Rn | AND register to A | 1 | 1 | 58-5F |
| ANL A,dir | AND direct byte to A | 2 | 1 | 55 |
| ANL A,@Ri | AND data memory to A | 1 | 1 | 56-57 |
| ANL A,#data | AND immediate to A | 2 | 1 | 54 |
| ANL dir,A | AND A to direct byte | 2 | 1 | 52 |
| ANL dir,#data | AND immediate data to direct byte | 3 | 2 | 53 |
| ORL A,Rn | OR register to A | 1 | 1 | 48-4F |
| ORL A,dir | OR direct byte to A | 2 | 1 | 45 |
| ORL A,@Ri | OR data memory to A | 1 | 1 | 46-47 |
| ORL A,#data | OR immediate to A | 2 | 1 | 44 |
| ORL dir,A | OR A to direct byte | 2 | 1 | 42 |
| ORL dir,#data | OR immediate data to direct byte | 3 | 2 | 43 |
| XRL A,Rn | Exclusive-OR register to A | 1 | 1 | 68-6F |
| XRL A,dir | Exclusive-OR direct byte to A | 2 | 1 | 65 |

| Mnemonic | Description | Bytes | Cycles | Hex Code |
|---|---|---|---|---|
| XRL A, @Ri | Exclusive-OR data memory to A | 1 | 1 | 66-67 |
| XRL A,#data | Exclusive-OR immediate to A | 2 | 1 | 64 |
| XRL dir,A | Exclusive-OR A to direct byte | 2 | 1 | 62 |
| XRL dir,#data | Exclusive-OR immediate to direct byte | 3 | 2 | 63 |
| CLR A | Clear A | 1 | 1 | E4 |
| CPL A | Complement A | 1 | 1 | F4 |
| SWAP A | Swap Nibbles of A | 1 | 1 | C4 |
| RL A | Rotate A left | 1 | 1 | 23 |
| RLC A | Rotate A left through carry | 1 | 1 | 33 |
| RR A | Rotate A right | 1 | 1 | 03 |
| RRC A | Rotate A right through carry | 1 | 1 | 13 |
| **DATA TRANSFER** | | | | |
| MOV A,Rn | Move register to A | 1 | 1 | E8-EF |
| MOV A,dir | Move direct byte to A | 2 | 1 | E5 |
| MOV A,@Ri | Move data memory to A | 1 | 1 | E6-E7 |
| MOV A,#data | Move immediate to A | 2 | 1 | 74 |
| MOV Rn,A | Move A to register | 1 | 1 | F8-FF |
| MOV Rn,dir | Move direct byte to register | 2 | 1 | A8-AF |
| MOV Rn,#data | Move immediate to register | 2 | 2 | 78-7F |
| MOV dir,A | Move A to direct byte | 2 | 1 | F5 |
| MOV dir,Rn | Move register to direct byte | 2 | 2 | 88-8F |
| MOV dir,dir | Move direct byte to direct byte | 3 | 2 | 85 |
| MOV dir,@Ri | Move data memory to direct byte | 2 | 2 | 86-87 |
| MOV dir,#data | Move immediate to direct byte | 3 | 2 | 75 |
| MOV @Ri,A | Move A to data memory | 1 | 1 | F6-F7 |
| MOV @Ri,dir | Move direct byte to data memory | 2 | 2 | A6-A7 |
| MOV @Ri,#data | Move immediate to data memory | 2 | 1 | 76-77 |
| MOV DPTR,#data | Move immediate to data pointer | 3 | 2 | 90 |
| MOVC A,@A+DPTR | Move code byte relative DPTR to A | 1 | 2 | 93 |
| MOVC A,@A+PC | Move code byte relative PC to A | 1 | 2 | 83 |
| MOVX A,@Ri | Move external data(A8) to A | 1 | 2 | E2-E3 |
| MOVX A,@DPTR | Move external data(A16) to A | 1 | 2 | E0 |
| MOVX @Ri,A | Move A to external data(A8) | 1 | 2 | F2-F3 |
| MOVX @DPTR,A | Move A to external data(A16) | 1 | 2 | F0 |
| PUSH dir | Push direct byte onto stack | 2 | 2 | C0 |
| POP dir | Pop direct byte from stack | 2 | 2 | D0 |
| XCH A,Rn | Exchange A and register | 1 | 1 | C8-CF |
| XCH A,dir | Exchange A and direct byte | 2 | 1 | C5 |
| XCH A,@Ri | Exchange A and data memory | 1 | 1 | C6-C7 |
| XCHD A,@Ri | Exchange A and data memory nibble | 1 | 1 | D6-D7 |
| CLR C | Clear carry | 1 | 1 | C3 |
| CLR bit | Clear direct bit | 2 | 1 | C2 |
| SETB C | Set carry | 1 | 1 | D3 |
| **Mnemonic** | **Description** | **Bytes** | **Cycles** | **Hex Code** |

| | | | | |
|---|---|---|---|---|
| SETB bit | Set direct bit | 2 | 1 | D2 |
| CPL C | Complement carry | 1 | 1 | B3 |
| CPL bit | Complement direct bit | 2 | 1 | B2 |
| ANL C,bit | AND direct bit to carry | 2 | 2 | 82 |
| ANL C,/bit | AND direct bit inverse to carry | 2 | 2 | B0 |
| ORL C,bit | OR direct bit to carry | 2 | 2 | 72 |
| ORL C,/bit | OR direct bit inverse to carry | 2 | 2 | A0 |
| MOV C,bit | Move direct bit to carry | 2 | 1 | A2 |
| MOV bit,C | Move carry to direct bit | 2 | 2 | 92 |
| **BRANCHING** | | | | |
| ACALL addr 11 | Absolute jump to subroutine | 2 | 2 | 11- F1 |
| LCALL addr 16 | Long jump to subroutine | 3 | 2 | 12 |
| RET | Return from sub-routine | 1 | 2 | 22 |
| RETI | Return from interrupt | 1 | 2 | 32 |
| AJMP addr 11 | Absolute jump unconditional | 2 | 2 | 01- E1 |
| LJMP addr 16 | Long jump unconditional | 3 | 2 | 02 |
| SJMP rel | Short jump (relative address) | 2 | 2 | 80 |
| JC rel | Jump on carry = 1 | 2 | 2 | 40 |
| JNC rel | Jump on carry = 0 | 2 | 2 | 50 |
| JB bit,rel | Jump on direct bit = 1 | 3 | 2 | 20 |
| JNB bit,rel | Jump on direct bit = 0 | 3 | 2 | 30 |
| JBC bit,rel | Jump on direct bit = 1 and clear | 3 | 2 | 10 |
| JMP @A+DPTR | Jump indirect relative DPTR | 1 | 2 | 73 |
| JZ rel | Jump on accumulator = 0 | 2 | 2 | 60 |
| JNZ rel | Jump on accumulator != 0 | 2 | 2 | 70 |
| CJNE A,dir,rel | Compare A,direct jne relative | 3 | 2 | B5 |
| CJNE A,#d,rel | Compare A,immediate jne relative | 3 | 2 | B4 |
| CJNE Rn,#d,rel | Compare reg,immediate jne relative | 3 | 2 | B8-BF |
| CJNE @Ri,#d,rel | Compare Ind,immediate jne relative | 3 | 2 | B6-B7 |
| DJNZ Rn,rel | Decrement register, jnz relative | 2 | 2 | D8-DF |
| DJNZ dir,rel | Decrement direct byte, jnz relative | 3 | 2 | D5 |
| **MISCELLANEOUS** | | | | |
| NOP | No operation | 1 | 1 | 00 |

*Table 17 M8052 Instruction Set Summary*

In Table 17, a range of continuous hex opcodes is shown. These codes are used for 8 different registers. The register number applicable for each code is defined by the lowest three bits of the code. An example of a non-continuous block of codes is 11 F1. These codes are used for absolute jumps and calls where the top 3 bits of the code are used to store the top three bits of the destination address.

The CJNE instructions use the #d abbreviation for immediate data. All others use #data.

### OSCILLATOR CIRCUIT

The clock selector circuit (SELCLK pin) selects one divider to keep the PLSES frequency to 3.579545MHz, regardless of the crystal selected (14.31818MHz or 21.47727MHz). To the M8052 section, the frequency corresponds to the crystal selected.

| SELCLK pin | Description |
|---|---|
| 0 | Use a 21.47727 MHz crystal |
| 1 | Use a 14.31818 MHz crystal |

*Table 18 Correspondance Between SELCLK Pin and Oscillator Frequency*

## RESET

### Reset Synchronisation

The RESET* input pin is sampled in State 5 phase 2 of every machine cycle. When RST is sampled low, a device reset is performed. The port pins will go into a high impedance state as soon as the RESET* goes low. After RESET* is sampled high, it will take 1 or 2 clock cycles before ALE or PSEN are driven.

### Reset Summary

The RST input performs the following functions:

The PC (Program Counter) is set to zero.
The DPTR (Data Pointer) is set to zero.
The Stack Pointer is set to 07 hex.
Register Bank zero is selected.
Ports 0, 1, 2, and 3 are set to input mode.
Interrupts are disabled.
All interrupt priorities are set low.
The Timer/Counters are stopped and the Timer Flags are cleared.
The Accumulator and B registers are set to zero.
The UART is disabled.

# SPECIAL FEATURES

### Idle Mode

This mode is entered by setting PCON.0 high. Idle mode gates off the clock to the CPU, but not the clock to the Interrupt, Timer and Serial Port functions. Idle Mode is exited by asserting any enabled interrupt, restarting the CPU in the interrupt service routine. Alternatively a hardware reset may be performed by pulling the RESET* pin low for a minimum of 2 machine cycles.

### Timer 2 Baud Rates

The M8052 will not generate a UART baud rate from the internal timer when the value in RCAP2 is 65535 (FFFF hex).

# Appendix A

## References

Many references are made throughout this data sheet to the EIA 600 series of documents, which specify the CEBus standard. These documents are available through Global Engineering, and can be obtained as follows:

**On-line:**

http://www.cebus.org/global.html
global@ihs.com

**Off-line:**

Global Engineering Documents
15 Inverness Way East
Englewood, COLORADO, 80112 ,USA
Phone: 1-800-854-7179
Fax: 1-303-397-2740

# IMPORTANT NOTICE

DOMOSYS Corporation (DOMOSYS) reserves the right to make changes to its products or services without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

DOMOSYS warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with DOMOSYS' standard warranty. Testing and other quality control techniques are utilized to the extent DOMOSYS deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

DOMOSYS PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of DOMOSYS products in such applications is understood to be fully at the risk of the customer. Questions concerning potential risk applications should be directed to DOMOSYS. In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

DOMOSYS assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does DOMOSYS warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of DOMOSYS covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

## Change Tracking

The following changes have taken place in this document :

| Date | Responsible | Description of change | Revised page(s) |
|---|---|---|---|
| Feb. 28, 1998 | H. Dufour | The Power Down bit has been removed. PCON.1 Not implemented. | 37 |
| Feb. 28, 1998 | H. Dufour | The Power Down mode is not implemented: the Power Down Mode paragraph has been removed. | 51 |
| June 23, 1998 | P. Longtin | Replace 21.31818MHz by 21.4772MHz. | 4 |
| Sept 23, 1998 | A. Dunn | Removed the “needs external pull-up” comment from the pins P3.4, P3.6, and P3.7 | 3 |
| Jan 17, 1999 | A. Dunn | Replaced "power saving mode" with "15 I/O pins". Added information about pins P3.2, P3.4, P3.6, and P3.7. Added note about no Schmitt Trigger on RESET* pin. Changed information about port pins retaining state for 19 oscillator cycles after RESET* went low; port pins actually go into a high impedance state as soon as RESET* goes lows. | 1, 3, 12, 51 |
| Apr 7, 1999 | H. Dufour | Corrected typo AD for DA in table 16 | 46 |
| Apr 21, 1999 | A. Dunn | Removed blank $P_{DISS}$ specification, it was irrelevant. | 5 |
| Apr 23, 1999 | A. Dunn | Fixed microcotroller typo | 7 |
| May 12, 1999 | A. Dunn | Fixed Request next Rcv timestamp byte typo, should have been 01 hex to Ph_Request instead of A1 hex | 26 |
| Aug 12, 1999 | A. Dunn | Clarified description of Preamble Field Accepted XMIT state. | 28 |
| Oct. 12, 1999 | H. Dufour | Bytes used for XCH A,dir corrected from 1 to 2 | 52 |
| Oct. 25, 1999 | H. Dufour | Bytes used for DJNZ dir,rel corrected from 2 to 3 Hex Code for MOVC A,@A+PC corrected from 94 to 83 | 52,53 |
| March 2000 | A.G. | Correction to output voltage | 5 |


**Domosys Corporation**
**1995, rue Jean-Talon Sud**
**Suite 202**
**Sainte-Foy, Quebec**
**Canada G1N 4H9**
**Tel: (418) 681-8022**
**Fax: (418) 681-8015**
**e-mail: support@domosys.com**
**www.domosys.com**