aJ-80™ Real-time Low Power Java™ Processor

Overview

The aJile Systems aJ-80 is a single-chip Java processor that powers networked smart sensors. aJile's Java processor directly executes Java Virtual Machine™ (JVM) bytecodes, real-time Java threading primitives and a number of extended bytecodes for embedded operations. The native JVM bytecode implementation eliminates the typical interpreter or JIT software layers and provide the most optimal Java performance in both memory requirements and execution time. In addition, Java threading primitives (wait, yield, notify, monitor enter/exit) are implemented as extended bytecodes eliminating the need for a the traditional RTOS. The result is extremely low executive overhead with thread to thread context switch times of less than 1µsec. The aJ-80 features on-chip memory and all the I/O functions required for use in many real-time networked embedded applications. The powerful combination of direct JVM bytecode execution, direct multithreading support, and fully protected multiple JVM environments is ideal for efficient, safe, and robust Java execution. The aJ-80 is ideally suited for real-time networked smart sensors used in industrial control, automotive, home automation and consumer applications.

Features

JEM2 32-bit Direct Execution Java Processor Core

- Native JVM bytecode
- Extended bytecodes for I/O and threading support
- IEEE-754 floating-point arithmetic
- Writeable control store supports custom extended bytecodes

Native Java Threading Support

- Hard real-time, multi-threading kernel in hardware
- Threading operations are atomic including true Java synchronization
- Built-in deterministic scheduling queues
- Directly supports the Real Time Specification for Java (RTSJ)
- Thread to thread yield in less than 1µsec
- Eliminates traditional RTOS layer

Multiple JVM Manager (MJM™)

- Support two independent JVM's
- Brick wall time and space protection
- Support external memory protection

Internal 48KB RAM

- 32KB dedicated data memory
- 16KB microcode memory

Memory Controller

- 8-bit interface
- Six chip selects to support ROM, Flash, SRAM, and peripheral devices

Dual 16550 compatible UARTs

- 128-byte FIFO on Rx and Tx
- Support IrDA physical layer protocol

Three 16-bit Timers/Counters

- Flexible count control and counter I/O
- Pulse Width Modulation (PWM)
- Waveform measurement

Serial Peripheral Interface (SPI)

- Master/Slave operation
- Four external chip selects
- Programmable transfer length

General Purpose I/O Ports

- Twenty-two I/O pins
- I/O programmable on a per-bit basis
- Flexible interrupt generation

Phase Locked Loop (PLL) and Power Management

- Transparent CPU power down when the "run queue" is empty
- Individual peripherals can be deactivated when not in use
- Global clock disable with external wake-up pin

IEEE 1149.1 (JTAG) Interface

- Boundary scan
- Low-level debugger interface
- JPDA Java Debugger Interface

Designed for ultra-low-power operation

- Less than 1mW/MHz power consumption
- Fully static operation up to 66 MHz
- Implemented in 3.3V and 0.25µm CMOS process
- Core operates at 2.5V

Housed in 100-lead QFP package

System Development Support

The aJ-80 processor, bundled with Sun's Java 2 Micro Edition (J2ME) Connected Limited Device Configuration (CLDC) Java runtime system, optimizing application builder, debugging tools and evaluation board provides a complete solution for implementing real-time networked embedded Java applications. Using commercial Java IDEs, application developers can create standalone real-time Java applications totally in Java with the performance and memory efficiency of systems programmed in C and assembly.

An aJ-80 based system can be configured to execute in real-time and/or dynamic environments to support a wide range of applications. The dynamic runtime supports the CLDC Mobile Information Device Profile (MIDP) to allow Java "MIDlets" to be downloaded and executed dynamically in separate JVM environments. Multiple "MIDlets" can be run simultaneously under the control of the Java Applications Manager (JAM) that maintains memory and execution time allocations. aJ-80 can also be configured to execute both real-time and dynamic applications in deterministic time sliced schedule.

The primary components of the development and runtime environments are summarized as follows:

Optimizing Linker/Application Builder

- GUI based application build configuration and control tool - JEM Builder
- Utilizes standard JVM class files generated by commercial Java IDEs
- Statically resolves class files and eliminates unused methods and fields
- Performs bytecode optimizations
- Performs method substitutions (method invokes replaced by extended bytecodes)
- Builds boot tables, class initialization code, and assigns interrupt and trap handlers
- Configures JVM's and memory layout

Java Runtime System

- Java run-time environment based on a J2ME CLDC
- Includes networking classes, storage classes, and Java communications API
- Dynamic runtime includes the JAM (class loader, verifier, scheduler) and GC components
- Device drivers for integrated peripherals and generic physical device interfacing in Java

Application Debugging Tools

- Host-target communications via an IEEE 1149 (JTAG) interface using a simple inexpensive IEEE-1284 cable
- Host-based full featured low-level debugger - Charade (Target level debugger threads and routines are not required)
- Host-based JDI provided to interface to commercial JDI compliant source-level debuggers

Architecture Overview

The system-on-a-chip (SOC) architecture of the aJ-80 is shown in Figure 1. The aJ-80 uses dual internal bus architecture: processor bus and peripheral bus. To minimize bus loading and the associated power consumption the processor bus is limited to those devices that require high bandwidth (CPU, memory, and external interface). The peripheral bus provides access to on-chip peripherals and is isolated from the processor bus via the peripheral bridge. The external bus interface generates the address, data, and control signals to directly connect to most memory and peripheral devices.

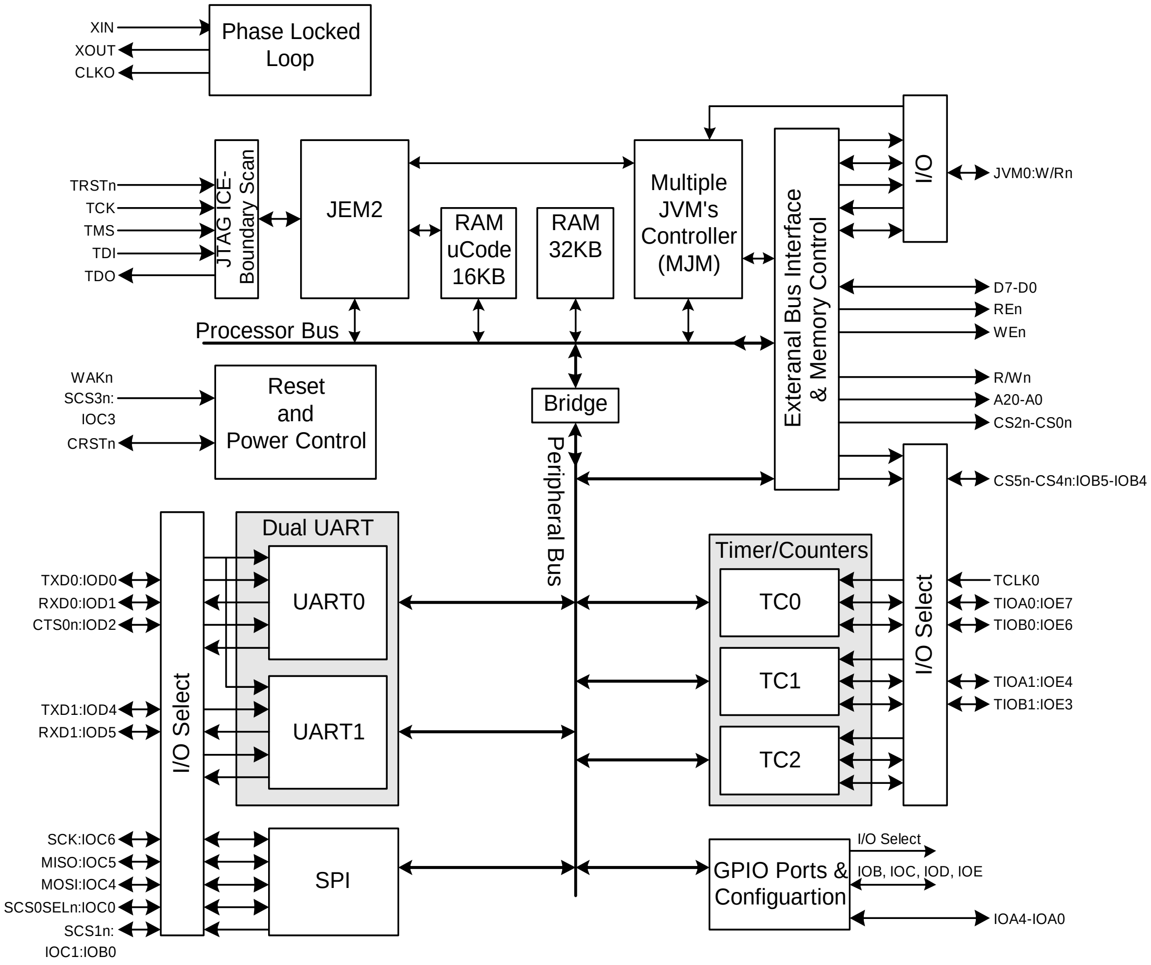

Figure 1: aJ-80 Architecture

Java Processing Core (JEM2)

The aJ-80 uses the JEM2 Java microprocessor to improve Java execution efficiency by eliminating the Java interpreter (software translation layer) and the RTOS kernel layer. Since JVM bytecodes are executed as native instructions the JEM2's Java performance is similar to RISC processors executing compiled C. The aJ-80 can be completely programmed in Java or in any language that compiles to Java class files. The direct implementation of real-time Java threading primitives results in extremely fast atomic threading operations (e.g., thread-to-thread yield is less than 1 microsecond), fast interrupt response, and deterministic scheduling (including true Java synchronization). Figure 2 illustrates the simplified block diagram of JEM2 core.

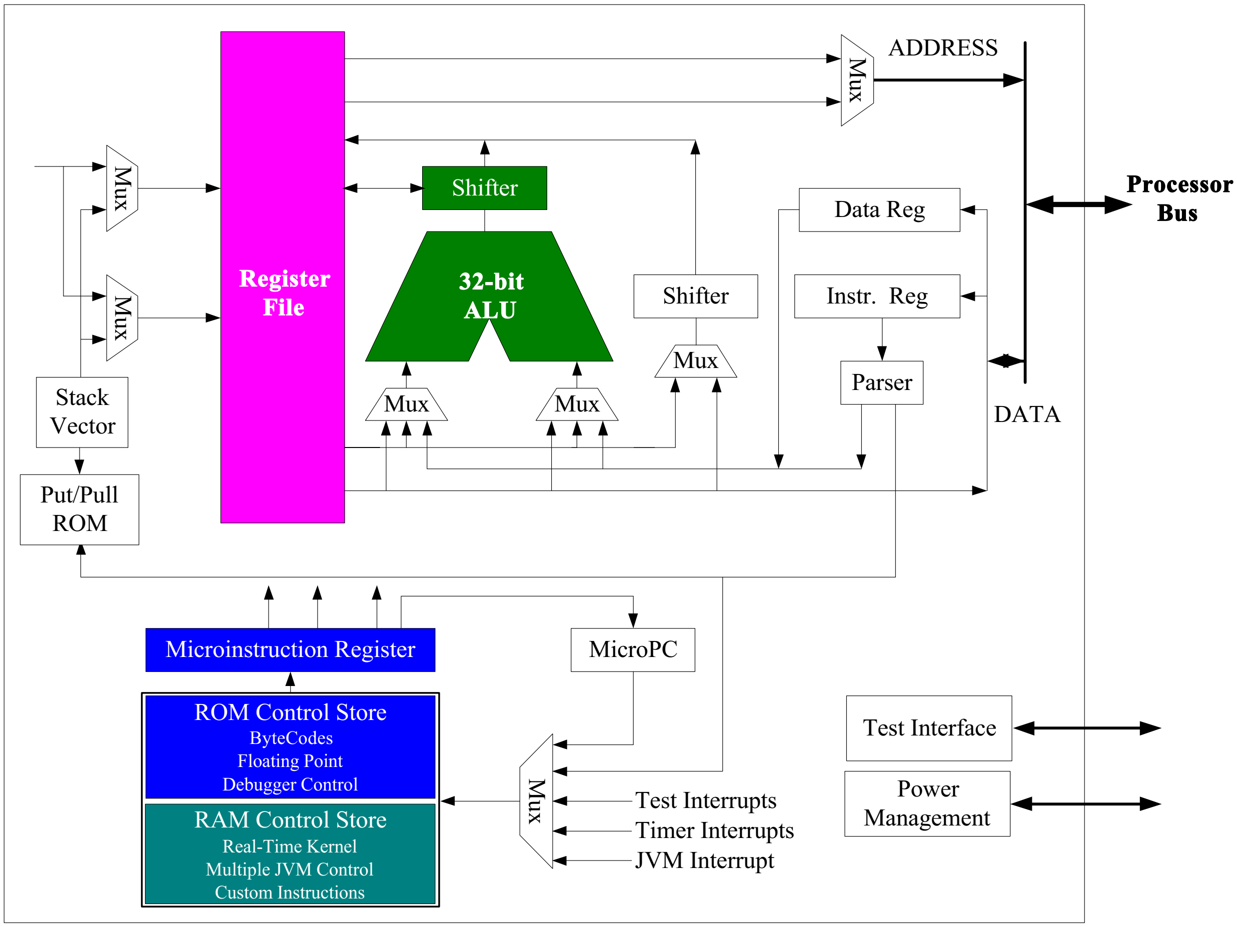

Figure 2: JEM2, Java Processing Core

Custom Instructions

The aJ-80 enables the use of custom microcode to implement new instructions. The new instructions can significantly increase the performance of frequently used algorithms. The power of custom instructions is reflected in the threading instructions of the JEM2. For example, the yield instruction results in a thread-to-thread switch of one microsecond while a typical RTOS written in a high level language that may take several milliseconds.

For an example of where a custom instruction could be used consider the square root method in java.lang.Math. This method could be implemented with microcode and initiated via an instruction. When processing a class file, the aJile tools will replace invokes of java.lang.Math.sqrt with the custom sqrt instruction. The performance advantage of a custom instruction varies from 5X for simple algorithms to 50X for complex algorithms.

Multiple JVM Manager (MJM)

The multiple JVM feature of the aJ-80 allows up to two independent Java applications to execute with a deterministic, time-sliced schedule and with full memory protection. Within its bounded execution interval and memory space, each JVM environment can employ its own multi-threading and memory utilization policies without threat of intervention by faulty or malicious applications.

The Multiple JVM Manger (MJM) provides timing resources and interrupt logic to ensure no JVM (applications) may interfere with the processing needs of the other JVM's. The MJM provides a timer to maintain the time slices allotted to each logical JVM. A separate timer is provided for each JVM (total of four) to schedule threads for that JVM. Figure 3 illustrates the MJM.

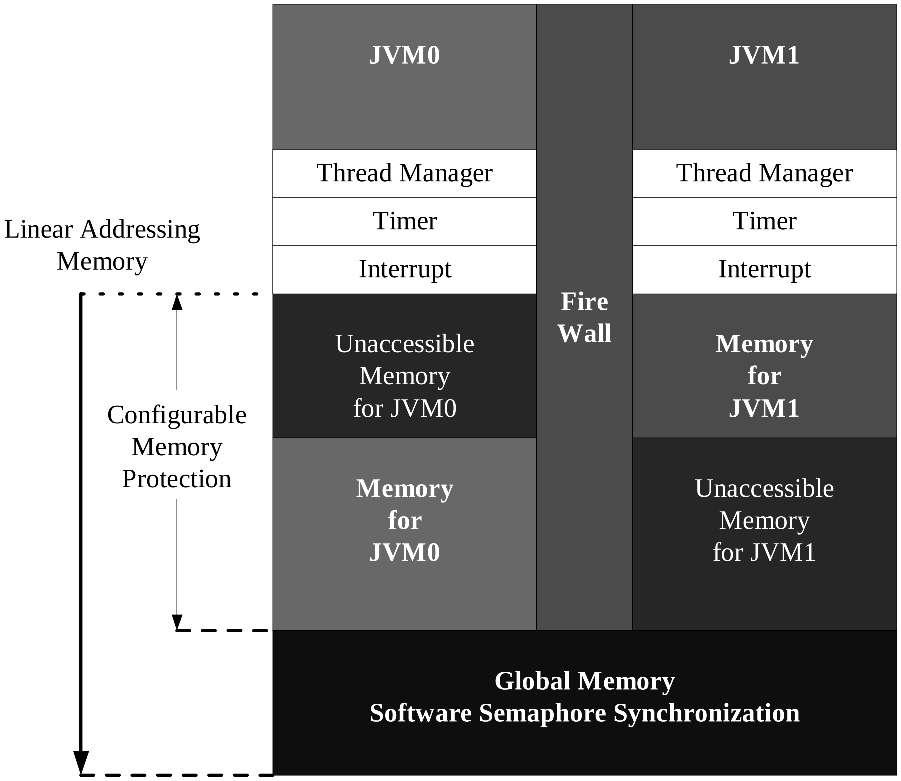

Figure 3: Multiple JVM Management (MJM)

Internal Memory

The aJ-80 provides 48 Kbytes of internal zero wait state memory. The 32 KBytes RAM is generally used for storage of the JEM2 processing stack. The remaining 16 Kbytes is used as microcode RAM for implementing the real-time kernel, extended bytecodes, and customized instructions.

External Bus Interface (EBI)

The external bus interface (EBI) generates the signals to control access to external memory and peripherals devices. The EBI can directly access up to 256 Mbytes of external memory. Additional address lines are accessible to further extend this memory space. The EBI provides eight chip selects. The EBI may be configured to support 32-bit, 16-bit, and 8-bit memory devices. Memory control signals are provided to enable direct connection to external memory and memory-mapped I/O devices. Transactions are controlled with the internal wait state generator with an external wait signal provided to extend access to slow devices. The EBI is illustrated in Figure 4.

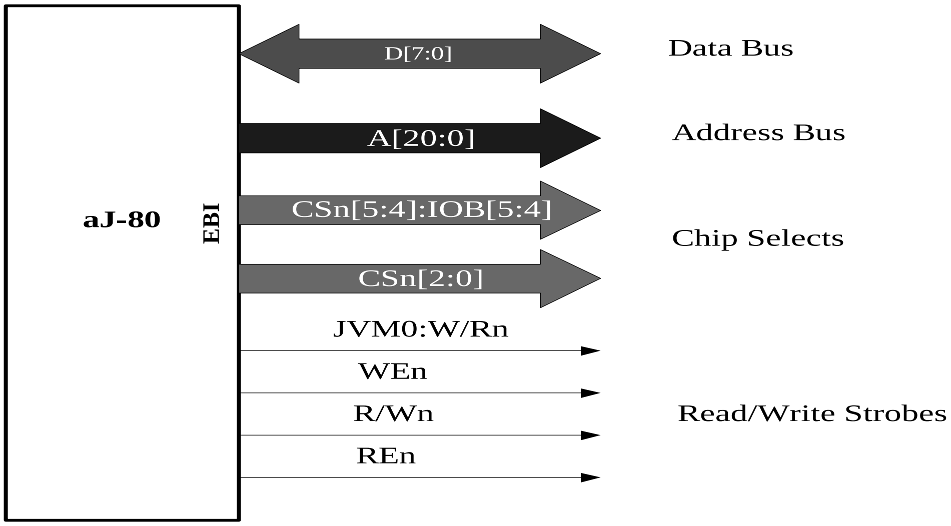

Figure 4: External Bus Interface

Bus Interface Timing

The aJ-80 provides a straightforward and flexible interface to external memory and peripheral devices. The aJ-80 performs the timing operations necessary for accessing external devices. Each chip select output has an associated configuration register to specify the setup times, hold times, wait states, and memory widths. The configuration registers are loaded as part of the reset initialization sequence. An external wait signal is provided to extend external transfers for slow devices or global busses. The fundamental write transfer is shown in figure 5 and the read transfer is shown in figure 6.

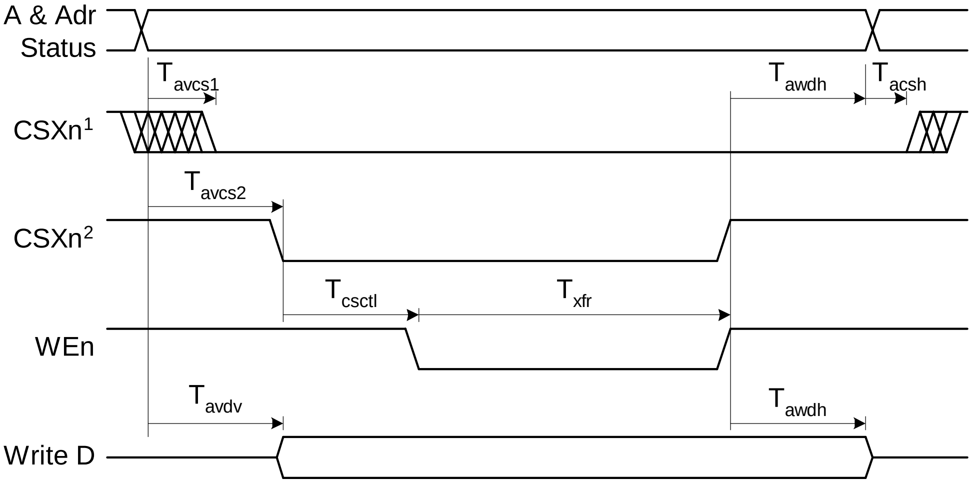

Figure 5: aJ-80 General Write Transfer

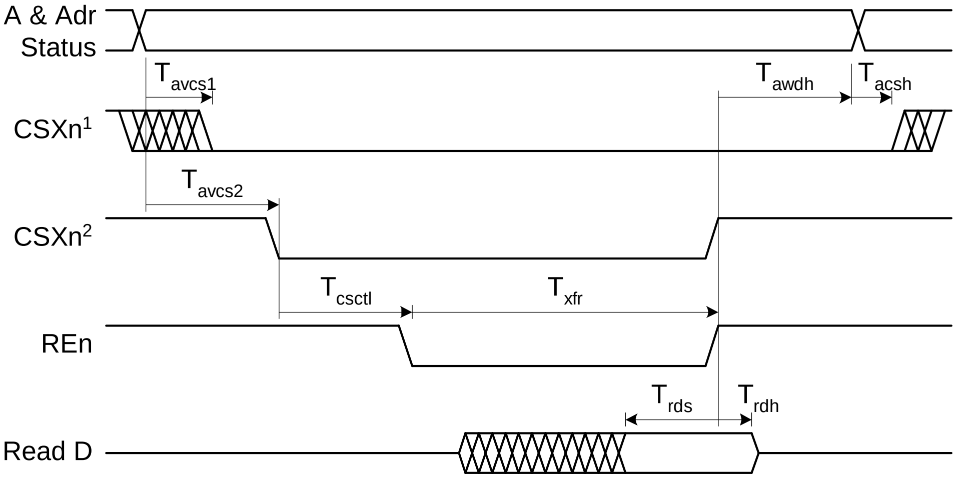

Figure 6: aJ-80 General Read Transfer Timing

Extended Transfer Timing

The aJ-80 allows bus interface allows transactions to be extended with the external transaction wait signal (WAITn). Extended cycles are useful for accesses to resources that have a varying response times (ex: a shared bus) or to slow devices

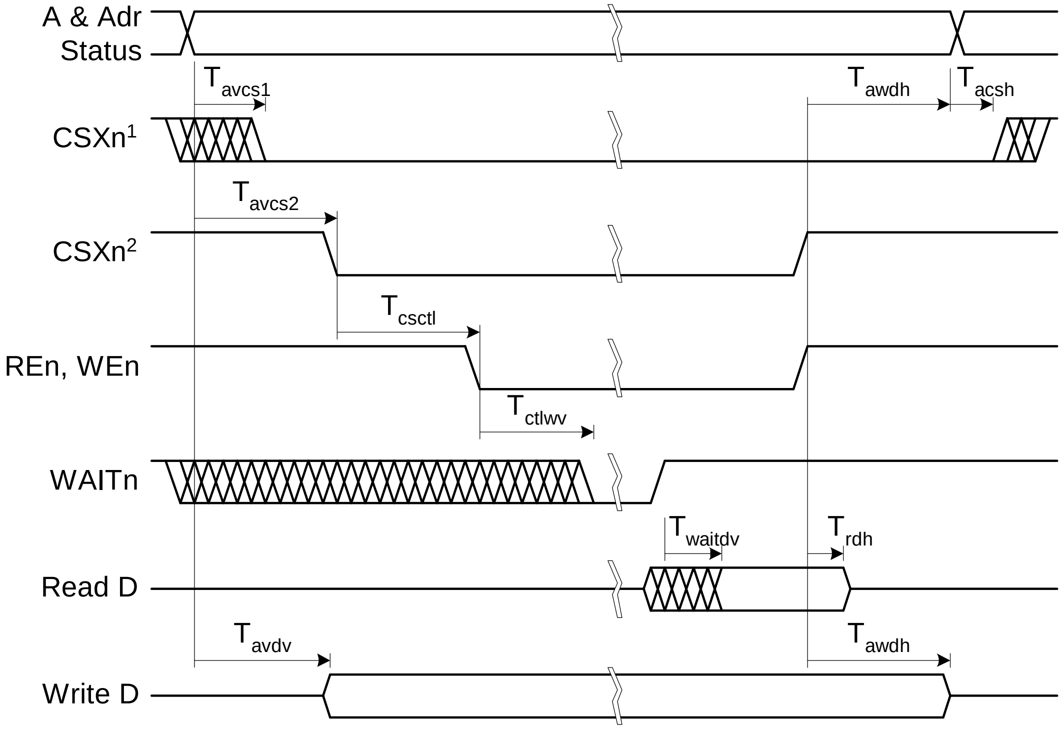

Figure 7: Extended Bus Transaction (WAITn)

Accessing Peripherals with ISA interfaces

Many peripherals have interfaces that support accesses on the ISA bus. To support devices with this type of interface the aJ-80 allows the CS4n and CS5n to operate as MEMRn and MEMWn signals (or IORn and IOWn). An access to an ISA oriented peripheral is illustrated in

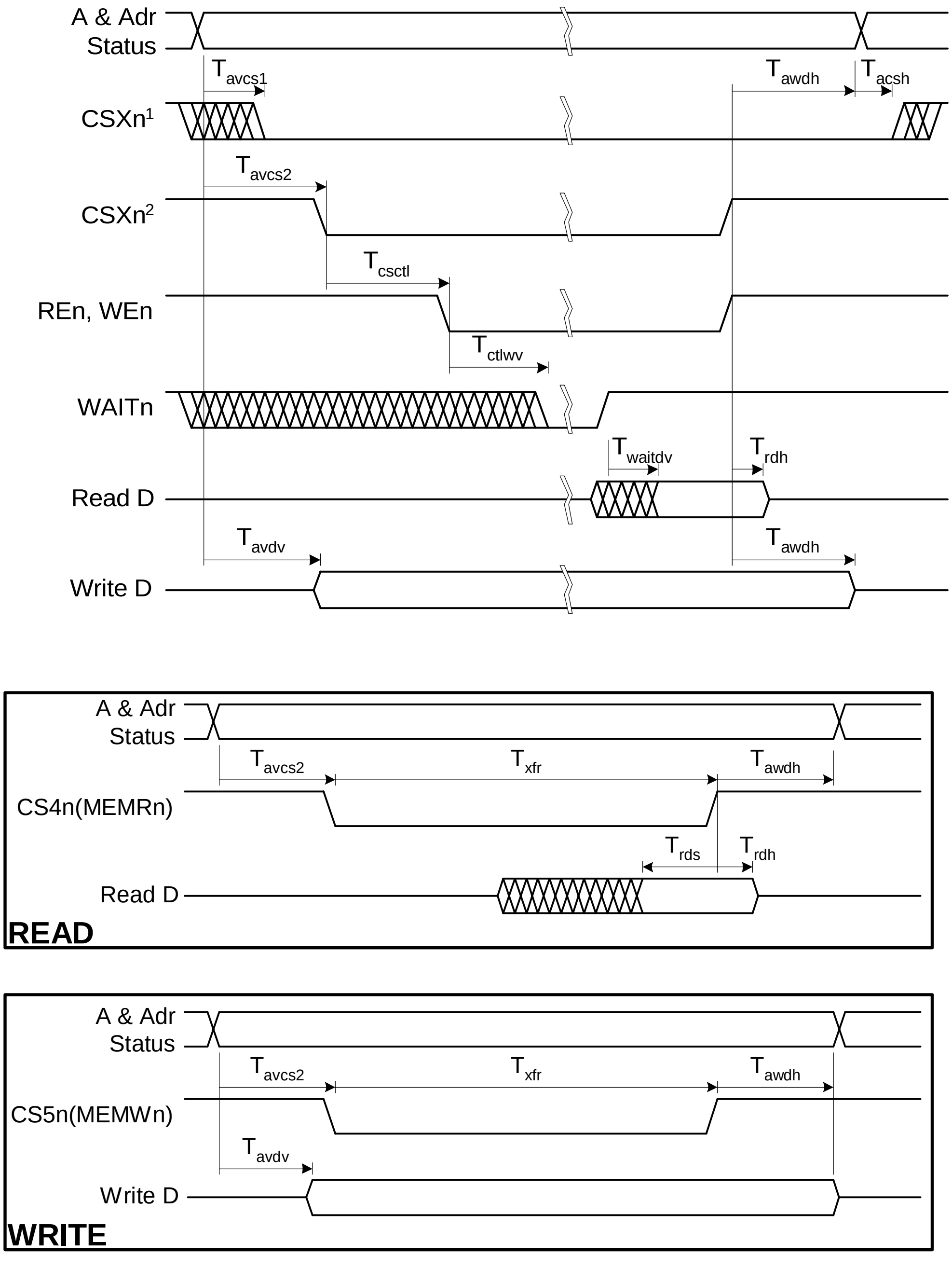

Figure 8: ISA-Oriented Peripheral Accesses.

Timer/Counter (TC)

The aJ-80 includes three 16-bit timer/counters (TC) that can perform a wide range of functions. The Timer/Counter functions include frequency measurement, event counting, interval measurement, delay timing, and pulse width modulation. The main features of the general purpose Timer/Counter are:

- Three 16-bit timers
- Internal chaining of timers
- External clock, triggering and gate control
- Two pulse width modulation and wave-form modules
- Flexible interrupt generation
- 16 bit prescaler

The Programmable Timer/Counter (TC) comprises a 16-bit prescaler and three 16–bit versatile timers/counter as shown in Figure 9. Two clock sources are provided as timing sources for the TC, the internal clock input and the external clock (TCLK0). The prescaler divides the selected input clock by PRL+1, where 0 ≤ PRL ≤ 65535, and provides the enable used by the versatile timers. The versatile timers may be employed as an interval or cyclic timer.

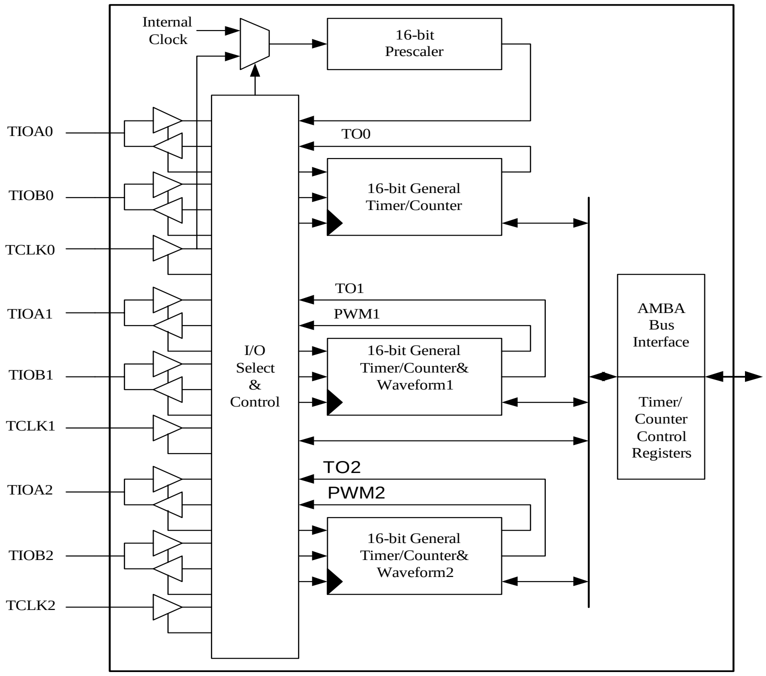
Internal Clock
16-bit Prescaler
TIOA0
TIOB0
TCLK0
TIOA1
TIOB1
TCLK1
TIOA2
TIOB2
TCLK2
I/O Select & Control
TO0
16-bit General Timer/Counter
TO1
PWM1
16-bit General Timer/Counter& Waveform1
TO2
PWM2
16-bit General Timer/Counter& Waveform2
AMBA Bus Interface
Timer/ Counter Control Registers

Figure 9: Timer/Counter Block diagram

Dual Universal Asynchronous Serial Port (DUART)

The aJ-80 provides two programmable UART channels. Each UART is compliant with the industry standard 16550 UART. The 128 byte FIFO minimizes the processor overhead required to communicate with the UART. The main features of the UART channels are:

- Software-compatible with NSC NS16550A
- Programmable word length, stop bits, and parity
- Programmable baud rate generator
- Interrupt generator
- Diagnostic loop-back mode
- 128-bytes FIFO on transmitter and receiver
- Independent modem line status
- IrDA 1.0 physical layer protocol with date rate up 115.2 kb/s

The UART is a universal asynchronous receiver/transmitter and is fully programmable through the peripheral bus. It supports programmable word length of five to eight bits, an optional parity bit, and one or two stop bits. If enabled, the parity can be odd, even or forced to a defined state. A 16-bit programmable baud rate generator and an 8-bit scratch register is included in the UART, together with two

128-byte FIFO's, one for transmitter and one for receiver. It has all the required modem control, interrupt handling, and error flags to indicate parity, overrun, and framing error conditions. An interrupt can be generated from any one of 10 sources. Figure 10 illustrates the simplified block diagram for UART

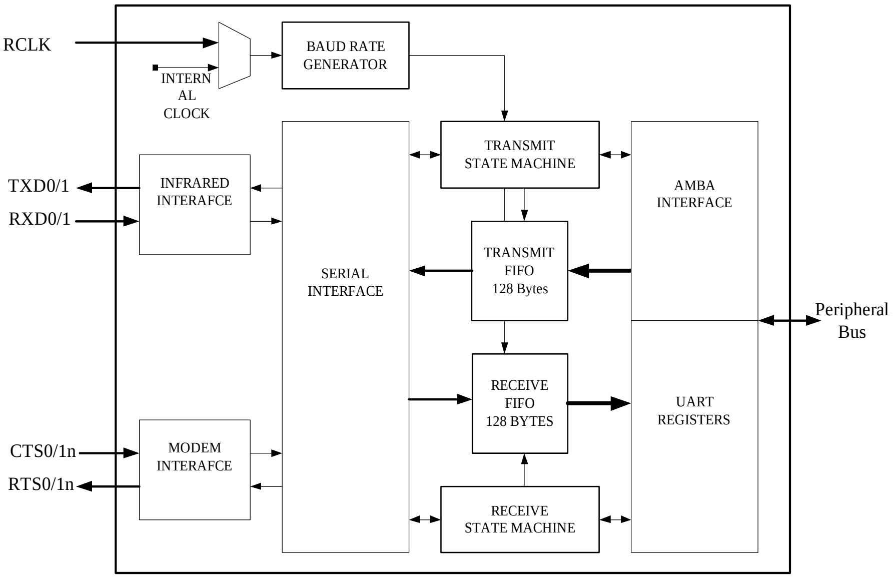

Figure 10: UART Block Diagram

The DUART als supports IrDA 1.0 physical layer protocol with data rates up to 115.2 kb/s. A block diagram of one end of an overall serial infrared link for data rates up to and including 115.2 kb/s is shown in Figure 11. The electrical pulses between the IR Transmit Encoder and the Output Driver & LED are 3/16 of a bit period in duration. The electrical pulses between the Detector & Receiver and the IR Receive Decoder are nominally of the same duration as those between the IR Transmit Encoder and the Output Driver & LED. Thus, the electrical signals at encoder interface of the Transducer are analogs of the optical signals at LEDs.

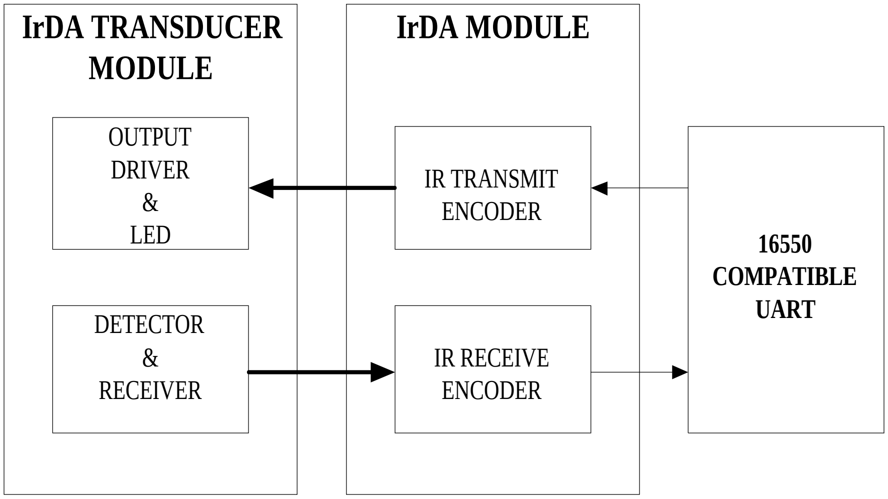

Figure 11: UART IRDA Block Diagram

General Purpose Input/Output (GPIO)

The aJ-80 includes twenty-two discrete I/O pins with five I/OA[4:0] connected to dedicated pins. The remaining I/O ports (IOB, IOC, IOD, and IOE) are multiplexed with the input/output signals from other resources on the aJ-80. Each I/O may be configured as an output, input, or bi-directional. Each GPIO input may be configured to generate a CPU interrupt on a high input, low input, rising edge, falling edge, or any signal transition. The dedicated port (IOA) uses 24mA output drivers and the shared GPIO ports uses 8mA drivers.

Serial Peripheral Interface (SPI)

The aJ-80 provides a SPI compatible synchronous serial interface which supports master/slave operation. A variety of low pin count peripheral devices are available with SPI interfaces. Such devices include A/D and D/A converters, serial EEPROMs, sensors, DSPs, UARTs, and CAN controllers. The aJ-80 directly supports up to four SPI slaves with support for additional slaves via GPIO pins or external decode logic. The SPI port supports programmable word length, clock rate, and clock polarity.

Test Interface

The aJ-80 communicates with software development environments via the standard IEEE 1149.1 (JTAG) port. The JTAG port consists of a five-wire interface, which is directly communicated with the internal Test Access Unit (TAU) to provide flexible user control over the operation of the aJ-80. This interface provides the fundamental target control features of run, step, halt, reset, memory access, register access, breakpoints, and software breakpoints. An external breakpoint input provides additional target control when used in conjunction with a logic analyzer.

Power Control

The aJ-80 provides two power savings modes. The “Idle” mode is essentially transparent to the executing software. Essentially the “idle” mode is entered when there is not a thread scheduled to run. The “Power-down” mode causes the aJ-80 to save its current operating state and disable its internal clock. These modes are further described in the following text. In addition to these two modes, each peripheral may be placed in a low-power mode ”standby” when not in use.

Idle mode

A feature of the built-in threading and multiple JVM architecture of the JEM cause the CPU to enter an “idle” state when there are no threads scheduled to run. During this mode the CPU will disable it’s internal clocking and thus enter a low power mode. Any unmasked interrupt (including the JVM switch) interrupt will immediately re-enable the CPU. The CPU will then determine which thread/JVM has requested execution. During the idle mode, enabled peripherals will continue to operate.

Power-down mode???

The JEM architecture has a built-in power-down mode. The power-down mode is initiated via the external power-down warning (PDWn) signal. The JEM2 power-down sequence causes each JVM to execute a “power-down” method. This enables each JVM to have specific power-down functionality. As a final action the internal global clock on the CPU and peripherals is deactivated. Power-down mode is intended to handle power transients while keeping memory alive. To resume normal operation, the power supply logic either signals the warm reset input (WRSTn) if memory has been maintained or the cold reset input (CRSTn) if not.

Standby mode

The aJ-80 can be put in “standby” mode to suspend execution and minimize power drain. Standby mode is initiated under executive software control via the “Standby” instruction to disable the internal clocking of the JEM and peripherals. The aJ-80 is “awakened” from the standby mode via the external WAKn pin. Exiting standby mode resumes normal execution immediately and eliminates the overhead of restarting the application. Note that memory must be kept alive during standby mode.

Phase Locked Loop (PLL)

The aJ-80 provides an internal Phase Locked Loop (PLL) to generate the high frequency clock for the JEM2 from a low-cost external oscillator reference. The PLL utilizes a 5 MHz to 50 MHz oscillator to generate internal clocks of up to 100MHz. The PLL may be disabled to provide power-savings for those systems that do not require the maximum performance from the JEM2.

Signal Description

Clock and reset signals

The aJ-80 utilizes a phase locked loop (PLL) to generate the high-speed internal clock from a low frequency external oscillator. The aJ-80 provides flexible reset control and a power down warning signal. The PLL and reset signals are described in the following table.

Table 1: Clock and Reset Signals

Signal Names	Types	Description
CLKO	O	*Clock Output*. This output clock is derived from the output of the internal PLL. The CLKO frequency is may be configured as 1/2, 1/4, or 1/8 of the CPU clock. The output may also be disabled to reduce power consumption and electromagnetic emission.
CRSTn	Open collector I/O	*Cold Reset*. The cold reset signal performs a power-on reset of the aJ-80. The CRSTn signal must be treated as an open collector signal. The aJ-80 has the capability to activate the CRSTn signal based on commands issued via the test interface. The test interface allows a software engineer to perform a system reset from the development environment.
WAKn SCS3n IOC	I O I/O	*Wakeup.* The wake-up input allows external logic to exit the JEM's standby mode. The JEM's idle mode is exited when an interrupt condition is present. *Slave Chip Select 3*. When operating in master mode the slave select signal may be used to select a SPI slave device. SCS3n is not used when the aJ-100 is operating as a SPI slave. SCSn1 is active low and may be put in to a high impedance state. *General Purpose I/O Port C, bit 3*. Refer to "General Purpose Input/Output" on page 18. Configuration of the pin operation is specified in the I/O configuration registers. The pin operates as IOC3 following a reset.
XIN	I	*Reference Oscillator*. This pin is used to provide the reference clock for the internal PLL. The range of the oscillator input is 5MHz-50MHz. The PLL multiplies the reference clock by 25 to drive the CPU at a maximum clock rate of 100MHz. The PLL may be bypassed to drive the CPU at lower clock rates. When bypassing the PLL (PLLBP low), XIN is used to directly drive the CPU.
XOUT	O	*Reference Oscillator Feedback*. XOUT is used to excite the reference oscillator circuit.

IEEE 1149.1 Test Interface (JTAG)

The aJ-80 provides an IEEE 1149.1 interface for performing the traditional board continuity tests and communicating with software development environment (SDE). This interface enables the SDE to provide user control over the operation of the CPU with a five-wire interface. The interface provides the fundamental target control features of run, step, halt, reset, memory access, register access, breakpoints, and software breakpoints. External breakpoint input provides additional target control when used in conjunction with a logic analyzer.

Table 2: JTAG Test Interface Signals

Signal Names	Types	Description
TCK	I	*Test Clock*. The test clock is the clock for 1149 test access port (TAP) controller, instruction register and all data registers.
TDI	I	*Test Data Input*. This pin is used to shift data into the 1194 data and instruction registers. TDI is captured on the rising edge of the TCK input.
TDO	O	*Test Data Output*. The pin is used to shift data out of the 1149 data and instruction registers. TDO changes on the falling edge of the TCK input.
TMS	I	*Test Mode Select*. The TMS input controls sequencing through the 1149 test access port (TAP) state machine.
TRSTn	I	*Test Reset*. Asserting the TRSTn input low, causes the test logic to be reset. TRSTn does not reset any system logic in the aJ-80. TRSTn resets the TAP controller and initializes the instruction register and some data registers to a known state.

Dual Universal Asynchronous Receiver/Transmitter (DUART)

The aJ-80 provides two UARTs that are compatible with the widely used 16550 architecture. The UARTs may utilize an external clock source or a clock derived from the internal CPU clock. An IrDA 1.0 encoder/decoder block is available for low cost wireless connectivity. The UART signals are described in the following table.

Table 3: DUART Signals

Signal Names	Types	Description
TXD0 IOD4	O I/O	*Transmit Data (UART channel 0)*. The transmit data output is used to send serial information from the aJ-80 to another serial device. TXD0 may be connected to the receiver of another UART, an RS-232 level translator, or an IrDA transmitter. *General Purpose I/O Port D, bit 4*. Refer to “General Purpose Input/Output” on page 18. Configuration of the pin operation is specified in the I/O configuration registers. The pin operates as IOD04 following a reset.
RXD0 IOD5	I I/O	*Receive Data (UART channel 0)*. The receive data input is used to send serial data from another serial device to the aJ-80. RXD0 may be connected to the transmitter of another UART, an RS-232 level translator, or an IrDA receiver. *General Purpose I/O Port D, bit 5*. Refer to “General Purpose Input/Output” on page 18. Configuration of the pin operation is specified in the I/O configuration registers. The pin operates as IOD5 following a reset.
CTS0n IOD6	I I/O	*Clear to Send (UART channel 0)*. A low value on CTS0n indicates that the modem or data set is ready to exchange data. The state of the CTSn0 signal is reflected in the modem status register. *General Purpose I/O Port D, bit 6*. Refer to “General Purpose Input/Output” on page 18. Configuration of the pin operation is specified in the I/O configuration registers. The pin operates as IOD6 following a reset.
TXD1 IOD0	O I/O	Transmit Data (UART channel 1). The transmit data output is used to send serial information from the aJ-80 to another serial device. TXD0 may be connected to the receiver of another UART, an RS-232 level translator, or an IrDA transmitter. General Purpose I/O Port D, bit 0. Refer to “General Purpose Input/Output” on page 18. Configuration of the pin operation is specified in the I/O configuration registers. The pin operates as IOD0 following a reset.
RXD1 IOD1	I I/O	*Receive Data (UART channel 1)*. The receive data input is used to send serial data from another serial device to the aJ-80. RXD0 may be connected to the transmitter of another UART, an RS-232 level translator, or an IrDA receiver. *General Purpose I/O Port D, bit 1*. Refer to “General Purpose Input/Output” on page 18. Configuration of the pin operation is specified in the I/O configuration registers. The pin operates as IOD1 following a reset.

Serial Peripheral Interface (SPI)

The SPI interface is used to communicate with a variety of low pin-count I/O devices and storage devices. Such devices include additional serial ports, CAN controllers, DSPs, A/D and D/A converters, sensors, and serial EEPROMs/PROMs. The SPI port interface signals are described in the following table.

Table 4: SPI Signals

Signal Names	**Types**	**Description**
MISO IOC5	I/O I/O	*Master In/Slave Out*. When operating in master mode MISO is input which receives serial data from the SPI slave. When operating in the slave mode the MISO is an output used to transmit data to the master device. *General Purpose I/O Port C, bit 5*. Refer to “General Purpose Input/Output” on page 18. Configuration of the pin operation is specified in the I/O configuration registers. The pin operates as IOC5 following a reset.
MOSI IOC4	I/O I/O	*Master Out/Slave In*. When operating in master mode MOSI is an output used to transmit data to the slave device. When operating in the slave mode MISO is input which receives data from the slave device. *General Purpose I/O Port C, bit 4*. Refer to “General Purpose Input/Output” on page 18. Configuration of the pin operation is specified in the I/O configuration registers. The pin operates as IOC4 following a reset.
SCK IOC6	I/O I/O	*SPI Transfer Clock*. The SPI clock controls the transfer of data between the aJ-80 and an external SPI device. SCK is driven by the aJ-80 when it is operating in the Master mode. When operating in the slave mode SCK is driven by an external device. The maximum SCK value is 1/16th of the internal aJ-80 CPU clock. *General Purpose I/O Port C, bit 6*. Refer to “General Purpose Input/Output” on page 18. Configuration of the pin operation is specified in the I/O configuration registers. The pin operates as IOC6 following a reset.
SCS0SELn IOC0	I/O I/O	*Slave Chip Select 0/Slave Mode Select*. When operating in master mode the SCS0SELn is output that may be used to select a SPI a slave device. When operating in the slave mode SCS0SELn is an input that is used to enable the SPI port for exchanging data. SCS0SELn is an active low signal. *General Purpose I/O Port C, bit 0*. Refer to “General Purpose Input/Output” on page 18. Configuration of the pin operation is specified in the I/O configuration registers. The pin operates as IOC0 following a reset.

Table 4: SPI Signals

Signal Names	Types	Description
SCS1n	O	*Slave Chip Select 1*. When operating in master mode the slave select signal may be used to select a SPI slave device. SCS1n is not used when the aJ-80 is operating as a SPI slave. SCSn1 is active low and may be put in to a high impedance state.
IOC1	I/O	*General Purpose I/O Port C, bit 1*. Refer to “General Purpose Input/Output” on page 18. Configuration of the pin operation is specified in the I/O configuration registers. The pin operates as IOC1 following a reset.
A24	I/O	*Address bit 24*. CPU address output and external master address input.
IOB0	I/O	*General Purpose I/O Port B, bit 0*. Configuration of the pin operation is specified in the I/O configuration registers. The pin operates as IOB0 following a reset.

General Purpose Input/Output

The aJ-80 has five 8-bit General Purpose Input/Output (GPIO) ports. Each GPIO pin may be individually configured as input or an output. Every GPIO pin may also be configured to generate a CPU interrupt. Interrupt flexibility is provided by allowing interrupts to be triggered on a rising edge, falling edge, either edge, high level, or low level. To minimize pin-count most of the GPIO signals are multiplexed with other I/O signals of the aJ-80. On a reset the multiplexed signals are configured as GPIO inputs. Operation of the multiplexed signals is controlled with the I/O configuration registers.

Table 5: GPIO Signals

Signal Names	**Types**	**Description**
IOA0	I/O	*General Purpose I/O port A, bit 0.* Bit programmable I/O signal and external interrupt input. (24 mA drive)
IOA1	I/O	*General Purpose I/O port A, bit 1.* Bit programmable I/O signal and external interrupt input. (24 mA drive)
IOA2	I/O	*General Purpose I/O port A, bit 2.* Bit programmable I/O signal and external interrupt input. (24 mA drive)
IOA3	I/O	*General Purpose I/O port A, bit 3.* Bit programmable I/O signal and external interrupt input. (24 mA drive)
IOA4	I/O	*General Purpose I/O port A, bit 4.* Bit programmable I/O signal and external interrupt input. (24 mA drive)
IOBx, IOCx, IODx, IOEx	I/O	*General Purpose I/O ports.* These seventeen general purpose I/O pins are shared with other pin functionality. Ex: IOE7 is shared with the general purpose timer 0 I/O A pin. (8 mA drive)

General Purpose Timer/Counters

Table 6: Timer / Counter Signals

Signal Names	Types	Description
TCK0	I	*External Clock (T/C0)*. TCK0 provides an optional external clock input for the General Purpose Timer/Counter 0 and the GPTC prescaler. Internal clock sources may also be selected as the clock source for Timer/Counter 0 or the prescaler.
TIOA0 IOE7	I/O I/O	*Input Control/Output A (T/C0)*. As input TIOA0 may be configured for a variety of control functions for Timer/Counter 0. TIOA0 may also be configured as a timer/counter output. *General Purpose I/O Port E, bit 7*. Refer to “General Purpose Input/Output” on page 18. Configuration of the pin operation is specified in the I/O configuration registers. The pin operates as IOE7 following a reset.
TIOB0 IOE6	I/O I/O	*Input Control/Output B (T/C0)*. As input TIOB0 may be configured for a variety of control functions for Timer/Counter 0. TIOB0 may also be configured as a timer/counter output. *General Purpose I/O Port E, bit 6*. Refer to “General Purpose Input/Output” on page 18. Configuration of the pin operation is specified in the I/O configuration registers. The pin operates as IOE6 following a reset.
TIOA1 IOE4	I/O I/O	*Input Control/Output A (T/C1)*. As input TIOA1 may be configured for a variety of control functions for Timer/Counter 1. TIOA1 may also be configured as a timer/counter output. *General Purpose I/O Port E, bit 4*. Refer to “General Purpose Input/Output” on page 18. Configuration of the pin operation is specified in the I/O configuration registers. The pin operates as IOE4 following a reset.
TIOB1 IOE3	I/O I/O	*Input Control/Output B (T/C1)*. As input TIOB1 may be configured for a variety of control functions for Timer/Counter 1. TIOB1 may also be configured as a timer/counter output. *General Purpose I/O Port E, bit 3*. Refer to “General Purpose Input/Output” on page 18. Configuration of the pin operation is specified in the I/O configuration registers. The pin operates as IOE3 following a reset.

External Bus Interface

Table 7: Bus Interface Signals

Signal Names	Types	Description
A0	I/O	*Address bit 0*. CPU address output and external master address input.
A1	I/O	*Address bit 1*. CPU address output and external master address input.
A2	I/O	*Address bit 2*. CPU address output and external master address input.
A3	I/O	*Address bit 3*. CPU address output and external master address input.
A4	I/O	*Address bit 4*. CPU address output and external master address input.
A5	I/O	*Address bit 5*. CPU address output and external master address input.

Table 7: Bus Interface Signals

Signal Names	Types	Description
A6	I/O	*Address bit 6*. CPU address output and external master address input.
A7	I/O	*Address bit 7*. CPU address output and external master address input.
A8	I/O	*Address bit 8*. CPU address output and external master address input.
A9	I/O	*Address bit 9*. CPU address output and external master address input.
A10	I/O	*Address bit 10*. CPU address output and external master address input.
A11	I/O	*Address bit 11*. CPU address output and external master address input.
A12	I/O	*Address bit 12*. CPU address output and external master address input.
A13	I/O	*Address bit 13*. CPU address output and external master address input.
A14	I/O	*Address bit 14*. CPU address output and external master address input.
A15	I/O	*Address bit 15*. CPU address output and external master address input.
A16	I/O	*Address bit 16*. CPU address output and external master address input.
A17	I/O	*Address bit 17*. CPU address output and external master address input.
A18	I/O	*Address bit 18*. CPU address output and external master address input.
A19	I/O	*Address bit 19*. CPU address output and external master address input.
A20	I/O	*Address bit 20*. CPU address output and external master address input.
CS0n	O	Chip Select #0. General purpose chip select for accessing the memory range 0x0000_0000..0x003F_FFFF. The CPU retrieves configuration information in the low portion of memory. Therefore, this select is typically connected to non-volatile memory device.
CS1n	O	*Chip Select #1.* General purpose chip select for accessing the memory range 0x0040_0000..0x007F_FFFF. This select is typically reserved for RAM.
CS2n	O	*Chip Select #2.* General purpose chip select for accessing the memory range 0x0080_0000..0x00BF_FFFF.
CS4n	O	*Chip Select #4*. CS4n operates as a general purpose chip select or as a read strobe for ISA oriented peripherals (IORn or MEMRn). The output operates as a general purpose chip select for the 0x0100_0000..0x013F_FFFF memory range. CS4n may also operate as the read strobe for an ISA oriented peripheral device. The output is activated during reads to the 0x0340_0000..0x037F_FFFF memory range. In this mode CS4n provides the read strobe (IORn or MEMRn) and CS5n provides the write strobe (IOWn or MEMWn) for the 0x0340_0000..0x037F_FFFF range.
IOB4	I/O	*General Purpose I/O Port B, bit 4*. Refer to “General Purpose Input/Output” on page 18. Configuration of the pin operation is specified in the I/O configuration registers. The pin operates as IOB4 following a reset.

Table 7: Bus Interface Signals

Signal Names	Types	Description
CS5n IOB5	O I/O	*Chip Select* #5. CS5n operates as a general purpose chip select or as a write strobe for ISA oriented peripherals (IOWn or MEMWn). The output operates as a general purpose chip select for the 0x0140_0000..0x017F_FFFF memory range. CS5n may also operate as the write strobe for an ISA oriented peripheral device. The output is activated during writes to the 0x0340_0000..0x037F_FFFF memory range. In this mode CS4n provides the read strobe (IORn or MEMRn) and CS5n provides the write strobe (IOWn or MEMWn) for the 0x0340_0000..0x037F_FFFF range. *General Purpose I/O Port B, bit 5*. Refer to “General Purpose Input/Output” on page 18. Configuration of the pin operation is specified in the I/O configuration registers. The pin operates as IOB5 following a reset.
D0	I/O	*Data bit 0*. CPU read or write data
D1	I/O	*Data bit 1*. CPU read or write data
D2	I/O	*Data bit 2*. CPU read or write data
D3	I/O	*Data bit 3*. CPU read or write data
D4	I/O	*Data bit 4*. CPU read or write data
D5	I/O	*Data bit 5*. CPU read or write data
D6	I/O	*Data bit 6*. CPU read or write data
D7	I/O	*Data bit 7*. CPU read or write data
JVM0 W/Rn	O O	*JVM Number bit 0*. The JVM output indicates which JVM is currently active. This may be used by external decode logic to determine if the specified JVM has privileges to access the specified address. *Write/Read Output.* A high level on W/Rn output indicates a write transaction while a low level indicates a read. The W/Rn signal is the inverse of R/Wn. Configuration of the pin operation is specified in the I/O configuration registers. The pin operates as W/Rn following a reset.
WAITn	I	*Wait Input.* The Wait signal may be used to indefinitely extend memory cycles. Use of the wait signal is specified in the configuration register of each chip select.
R/Wn	I/O	*Read/Write Output.* A high level on R/Wn output indicates a read transaction while a low level indicates a write. The R/Wn signal is the inverse of W/Rn.
REn	O	*Read Enable Strobe.* The read enable strobe is used to indicate the address and chip select signals are valid and a read operation is to proceed. The read enable output may be directly connected to the read enables or output enables of most memory devices.
WEn	O	*Write Enable.* The write enable strobe is used to indicate the address and chip select signals are valid. The write enable output may be directly connected to the write enable inputs on most volatile memory devices (RAM).

Power/Ground

Table 8: Power and Ground Signals

Signal Names	Types	Description
VDDIO	I	Nine connections to I/O Power, 3.3V
VSSIO	I	Nine connections to I/O Ground
VDDCORE	I	Four connections to Core Power,2.5V
VSSCORE	I	Four connections to Core Ground
PLLAVDD	I	PLL Analog Power, 2.5V
PLLAVSS	I	PLL Analog Ground
PLLDVDD	I	PLL Digital, 2.5V
PLLDVSS	I	PLL Ground
PLLGRVDD	I	PLL Guard Ring Power, 2.5V
PLLGRVSS	I	PLL Guard Ring Ground

DC Characteristics

This section contains the electrical characteristics for aJ-80 microcontroller.

Table 9: Maximum Ratings

Rating	Value
Maximum supply voltage for I/O	4.6V
Maximum supply voltage for Core	3.5V
Voltage on any input with respect to VSS	-0.5V to 6.0V
Voltage on any output with respect to VSS	-0.5V to 4.6V
Operating temperature range	0C to 70C
Storage temperature	-40C to 150C

Table 10: DC Characteristics

Symbol	Characteristics	Condition	Min	Typ	Max	Unit
VDDIO	Supply I/O Voltage		3.0	3.3	3.6	V
VDDCORE	Supply Core Voltage		2.25	2.5	2.75	V
I_{CC}	Operating current at 100 MHz		TBD	TBD	TBD	mA
I_{STB}	Standby current		TBD	TBD	TBD	µA
V_{IH}	Input high voltage (LVTTL)		2.3V		5.5V	V
V_{IL}	Input low voltage (LVTTL)		-0.5		1.0	V
V_{OH}	Output high voltage (standard drive) High drive (GPIO port A, IOAx)	I_{OH}=8.0mA I_{OH}=24.0mA	2.4 2.4			V V
V_{OL}	Output low voltage (standard drive) High drive (GPIO port A, IOAx)	I_{OL}=8.0mA I_{OL}=24.0mA			0.4 0.4	V V
I_{IL}	Input leakage current (no pull-up or pull-down)	V_{IN}= 3.3 or 0V	±10nA		±10µA	
I_{OZ}	Tristate output leakage current (no pull-up or pull-down)	V_{OUT}= 3.3 or 0V	±10nA		±10µA	
I_{OH}	Output high current (V_{OH}=2.4V) (standard drive) High drive (GPIO port A, IOAx)		8 24			mA mA
I_{OL}	Output low current (V_{OL}=0.4V) (standard drive) High drive (GPIO port A, IOAx)		8 24			mA mA
R_{PU}	Pull-up resistor		70K	98K	172K	Ohm

Pin ListingaJ-80 Pin Assignment

Table 11: Pin Listing

Pin #	Signal Name	Type	I/O and Output Reset State
1	TDO	0 (with pull-up)	Pull-up
2	VDDIO	I/O power	
3	VSSIO	I/O ground	
4	TIOB1:IOE3	I/O (with pull-up)	Pull-up
5	TIOA1:IOE4	I/O (with pull-up)	Pull-up
6	TIOB0:IOE6	I/O (with pull-up)	Pull-up
7	TIOA0:IOE7	I/O (with pull-up)	Pull-up
8	TCLK0	I (with pull-up)	Pull up
9	VDDCORE	I/O power	
10	VSSCORE	I/O ground	
11	CS0n	O	High
12	CS1n	O	High
13	CS2n	O	High
14	D0	I/O	Tri-state
15	A0	I/O	Tri-state
16	D1	I/O	Tri-state
17	A1	I/O	Tri-state
18	VSSIO	I/O ground	
19	VDDIO	I/O power	
20	D2	I/O	Tri-state
21	A2	I/O	Tri-state
22	D3	I/O	Tri-state
23	A3	I/O	Try-state
24	D4	I/O	Tri-state
25	A4	I/O	Tri-state
26	VDDIO	I/O power	
27	VSSIO	I/O ground	
28	D5	I/O	Tri-state
29	A5	I/O	Tri-state
30	D6	I/O	Tri-state
31	A6	I/O	Tri-state
32	D7	I/O	Tri-state
33	A7	I/O	Tri-state
34	WAITn	I	
35	R/Wn	I/O	Tri-state
36	WEn (with pull-up)	O	Pull-up
37	REn (with pull-up)	O	Pull-up
38	VSSCORE	I/O ground	
39	VDDCORE	I/O power	
40	A8	I/O	Tri-state
41	VSSIO	I/O ground	
42	VDDIO	I/O power	
43	A9	I/O	Tri-state

Table 11: Pin Listing

Pin #	Signal Name	Type	I/O and Output Reset State
44	A10	I/O	Tri-state
45	A11	I/O	Tri-state
46	A12	I/O	Tri-stage
47	A13	I/O	Tri-state
48	A14	I/O	Tri-state
49	A15	I/O	Tri-state
50	VSSIO	I/O ground	
51	VDDIO	I/O power	
52	IOA0	I/O (with pull-up)	Pull-up
53	IOA1	I/O (with pull-up)	Pull-up
54	IOA2	I/O (with pull-up)	Pull-up
55	IOA3	I/O (with pull-up)	Pull-up
56	A16	I/O (with pull-up)	Pull-up
57	VDDIO	I/O power	
58	VSSIO	I/O ground	
59	IOA4	I/O (with pull-up)	Pull-up
60	CS5n:IOB5	I/O (with pull-up)	Pull-up
61	CS4n:IOB4	I/O (with pull-up)	Pull-up
62	VSSCORE	Core ground	
63	VDDCORE	Core power	
64	SCS0SELn:IOC0	I/O (with pull-up)	Pull-up
65	TMS	I (with pull-up)	Pull-up
66	MOSI:IOC4	I/O (with pull-up)	Pull-up
67	MISO:IOC5	I/O (with pull-up)	Pull-up
68	SCK:IOC6	I/O (with pull-up)	Pull-up
69	RXD1:IOD1	I/O (with pull-up)	Pull-up
70	TXD1:IOD0	I/O (with pull-up)	Pull-up
71	VSSIO	I/O ground	
72	VDDIO	I/O power	
73	CTS0n:IOD6	I/O (with pull-up)	Pull-up
74	RXD0:IOD5	I/O (with pull-up)	Pull-up
75	TXD0:IOD4	I/O (with pull-up)	Pull-up
76	PLLGRVSS	PLL ground	
77	PLLGRVDD	PLL power	
78	PLLAVSS	PLL analog ground	
79	PLLAVDD	PLL analog power	
80	PLLDVSS	PLL digital ground	
81	PLLDVDD	PLL digital power	
82	VSSIO	I/O ground	
83	VDDIO	I/O power	
84	XIN	I	
85	XOUT	O	
86	CLKO	O	Low
87	WAKn:SCS3n:IOC3	I/O (with pull-up)	Pull-up
88	CRTSn	I/O (with pull-up)	Low

Table 11: Pin Listing

Pin #	Signal Name	Type	I/O and Output Reset State
89	A17	I/O (with pull-up)	Pull-up
90	VDDCORE	Core power	
91	VSSCORE	Core ground	
92	A18	I/O (with pull-up)	Pull-up
93	A19	I/O (with pull-up)	Pull-up
94	A20	I/O (with pull-up)	Pull-up
95	TCK	I (with pull-up)	Pull-up
96	VSSIO	IO ground	
97	VDDIO	I/O power	
98	TRSTn	I (with pull-up)	
99	SCS1n:IOC1/A24: IOB0	I/O	
100	TDI	I (with pull-up)	Pull-up

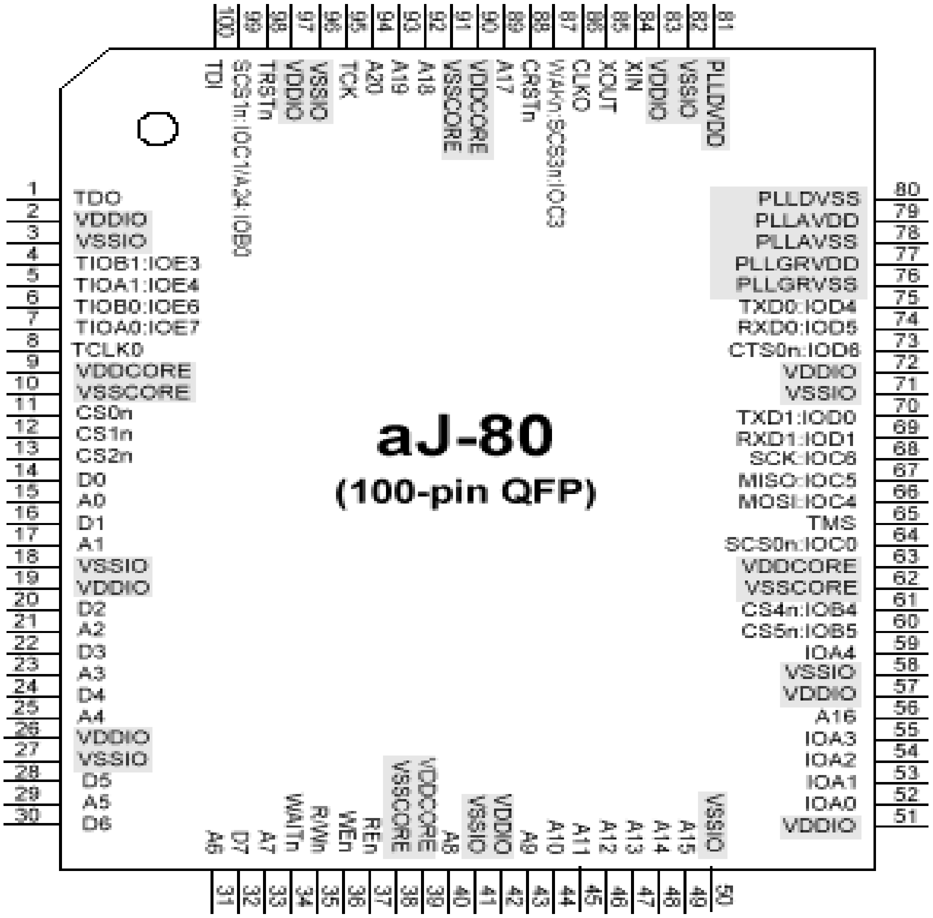

Figure 12: aJ-80 Pin Assignment

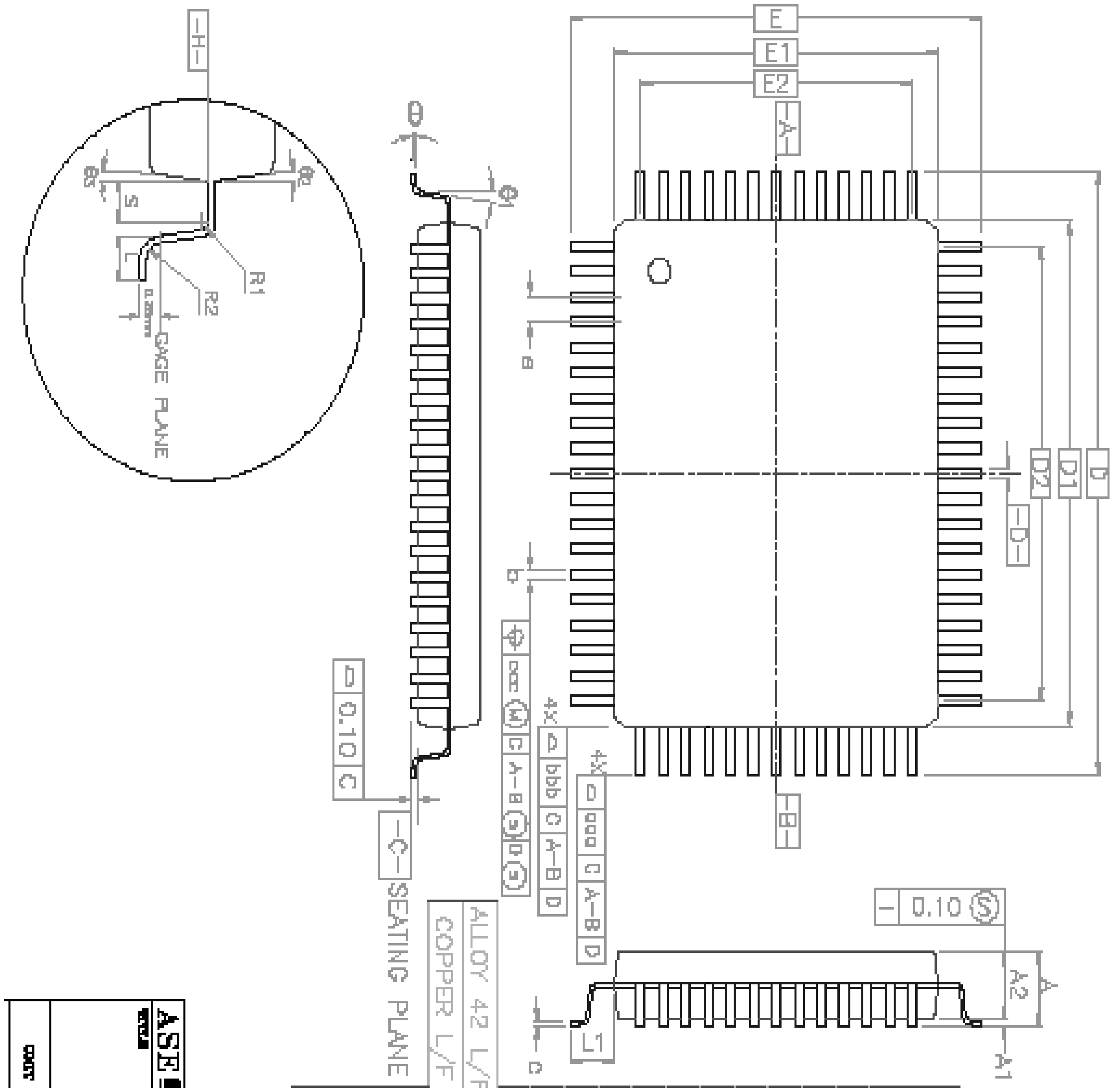

Figure 13: Package Outline Drawing

Symbol	Millimeters			Inches		
	Minimum	Nominal	Maximum	Minimum	Nominal	Maximum
A	-	-	3.40	-	-	0.134
A_1	0.15	0.25	0.35	0.006	0.010	0.014
A_2	2.55	2.72	3.05	0.100	0.107	0.120
D	23.90 BASIC			0.941 BASIC		
D_1	20.00 BASIC.			0.787 BASIC		
E	17.90 BASIC			0.705 BASIC		
E_1	14.00 BASIC			0.551 BASIC.		

Symbol	Millimeters			Inches		
	Minimum	Nominal	Maximum	Minimum	Nominal	Maximum
R_2	0.13	-	0.30	0.005	-	0.012
R_1	0.13	-	-	0.005	-	-
θ	0°	3.5°	7°	0°	3.5°	7°
θ_1	0°	-	-	0°	-	-
Alloy 42 θ_2, θ_3	7°			7°		
C	0.11	0.15	0.23	0.004	0.006	0.009
L	0.73	0.88	1.03	0.029	0.035	0.041
S	0.4	-	-	0.016	-	-
b	0.22	0.30	0.38	0.009	0.012	0.015
e	0.65BSC			0.026BSC		
D2	18.85REF			0.742		
E2	12.35REF			0.486		
Tolerance of form and position						
aaa	0.25			0.010		
bbb	0.20			0.008		
ccc	0.12			0.005		