Wireless LAN Access Point Controller

The Intersil ISL3856 Wireless LAN Access Point Controller is part of Intersil's Wireless LAN chip sets targeting Access Point applications.

The ISL3856 Access Point Controller is an ARM940 core controller with an onboard MAC to Ethernet (10/100 Base T) interface. The ISL3856 directly interfaces with the Intersil HFA386x family of Baseband Processors, offering a complete end-to-end IEEE 802.11b compliant chip set solution for wireless LAN products. Protocol and PHY support are implemented in firmware to allow custom protocol and different PHY transceivers.

The ISL3856 is a Harvard architecture cached processor. The separate instruction and data caches in this design are 4kbytes each in size with a four-word line length. A protection unit allows the memory to be segmented and protected in a simple manner. There is no virtual physical address mapping. Write-back cache schemes and write buffers are used to optimize performance and minimize bus traffic thus reducing system power consumption. This Processor Core is implemented using a five-stage pipeline consisting of fetch, decode, execute, memory and write stages.

Firmware implements the full IEEE 802.11b Wireless LAN MAC protocol. It supports Infrastructure mode BSS operation under DCF, and operation under the optional Point Coordination Function (PCF). All low-level 802.11 functions are handled by firmware. Additional firmware functions specific to access point applications are also available.

The ISL3856 is the industry's first Access Point on a chip, which implements both the IEEE 802.11 MAC protocol and the MAC bridging function, which in alternative solutions requires a separate external processor. For network management, an SNMP agent is implemented for access to the MIB.

Designing wireless protocol systems using the ISL3856 is made easier with Intersil supplied firmware, software device drivers, and complete documentation.

Ordering Information

PART NUMBER	TEMP. RANGE (^{o}C)	PACKAGE	PKG. NO.
ISL3856CK	0 to 70	BGA 11x11	V169.11x11
ISL3856CK-T	0 to 70	Tape and Reel 1000 Per Reel	

Features

- ARM940T Core
- Baseband Processor Interface Providing a Direct Transmit and Receive Serial Interface to an External Baseband Processor
- Serial Control Port (SCP), Supporting Serial Communication for Control of External Devices
- Memory Interfaces Supporting SRAM, ROM and SDRAM Memories, at 8, 16 or 32 bits
- Real Time Clock
- MII Interface
- 3 General Purpose Timers
- Interrupt Controller
- 16 General Purpose I/Os (GPIO)
- A UART to Enable System Debugging, Multiplexed onto GPIO Lines
- On-Chip PLL for Clock Generation
- Test Interface Controller (TIC) to Support Manufacturing Tests
- JTAG Interface for Boundary Scan and Debug Port
- Power Management Capabilities
- IEEE802.11 Standard Data Rates: 1, 2, 5.5 and 11Mbps
- Part of the Intersil PRISM Wireless LAN Chip Set
- Full Implementation of the MAC Protocol Specified in IEEE Standards 802.11-1999 and 802.11b
- Operation at 3.3V Supply
- 169 Pin BGA Package Targeted for Compact Gateways

Applications

- High Data Rate Wireless LAN up to 54Mbits
- Residential Gateways
- Wireless LAN Modules such as Wireless Gateways
- Wireless LAN Access Points
- Wireless Bridge Products
- Wireless Point-to-Multipoint Systems

ISL3856 Pin Number Assignments

PIN NUMBER	SIGNAL NAME	PIN NUMBER	SIGNAL NAME	PIN NUMBER	SIGNAL NAME	PIN NUMBER	SIGNAL NAME
A1	SCFRM(O)	B1	SCFRM(2)	C1	GPIO(0) (PA_PE)	D1	GPIO(4) (TX_PE)
A2	MIA(17)	B2	SCCLK	C2	SCTXD	D2	GPIO(1) (MD_RDY)
A3	MIA(13)	B3	MIA(14)	C3	MIA(16)	D3	SCRXD
A4	MIA(10)	B4	MIA(11)	C4	MIA(15)	D4	SCFRM(1)
A5	MIA(7)	B5	V_{SS}	C5	MIA(12)	D5	V_{DD}
A6	MIA(3)	B6	MIA(2)	C6	MIA(6)	D6	MIA(9)
A7	V_{DD}	B7	MIWE-	C7	MIA(4)	D7	MIA(5)
A8	MIA(0)	B8	MIRDY	C8	MIA(1)	D8	V_{SS}
A9	MIOE-	B9	MIBLS(2)-	C9	MIBLS(1)-	D9	MIBLS(3)-
A10	V_{DD}	B10	MIAA	C10	MISA	D10	MICLK(1)
A11	MICLK(3)	B11	MICLK(2)	C11	MICS(3)-	D11	MID15
A12	MICLK(0)	B12	MICS(2)-	C12	MID(14)	D12	MID(11)
A13	V_{DD}	B13	MICS(1)-	C13	V_{SS}	D13	MID(10)
E1	GPIO7 (MAN_RESET)	F1	GPIO10 (MICLKEN1)	G1	GPIO13 (CAL_EN)	H1	GPIO14 (CCA)
E2	V_{SS}	F2	V_{DD}	G2	GPIO8(TR_SW)	H2	GPIO12 (PE2)
E3	GPIO5 (MDC)	F3	GPIO9 (T/R_SW_BAR)	G3	V_{SS}	H3	nTRST
E4	GPIO6 (MDIO)	F4	GPIO11 (PE1)	G4	GPIO15 (RESOUT)	H4	TDI
E5	GPIO2 (Radio_PD)	F5	GPIO3 (LED_ETHER)	G5	V_{SS}	H5	TDO
E6	MIA(8)	F6	V_{DD}	G6	V_{SS}	H6	BPTXACT
E7	V_{SS}	F7	V_{SS}	G7	V_{SS}	H7	V_{SS}
E8	MIBLS(O)-	F8	MID(13)	G8	V_{SS}	H8	NC
E9	V_{SS}	F9	MID(9)	G9	V_{SS}	H9	MID(25)
E10	MICS(O)-	F10	V_{DD}	G10	MID(5)	H10	MID(30)
E11	MID(12)	F11	MID(6)	G11	MID(4)	H11	MID(1)
E12	MID(8)	F12	MID(2)	G12	V_{DD}	H12	MID(31)
E13	MID(7)	F13	V_{SS}	G13	MID(3)	H13	MID(0)
J1	TMS	K1	TST(0)	L1	BPTXD	M1	BPRXD
J2	TCLK	K2	TST(1)	L2	BPTXCLK	M2	BPRXCLK
J3	V_{DD}	K3	BPTXRDY	L3	BPRXRDY	M3	ENRXD0
J4	BPRXACT	K4	ENCRSDV	L4	ENTXEN	M4	ENTXD(0)
J5	ENRXD(1)	K5	ENTXER	L5	ENTXD(2)	M5	ENTXD(3)
J6	V_{DD}	K6	V_{SS}	L6	ENTXCLK	M6	NC
J7	V_{SS}	K7	V_{SSA}	L7	PMXOUT	M7	NC
J8	PMRES	K8	NC	L8	ENRXD(2)	M8	ENRXDV
J9	MID(21)	K9	NC	L9	NC	M9	ENCOL

ISL3856 Pin Number Assignments (Continued)

PIN NUMBER	SIGNAL NAME	PIN NUMBER	SIGNAL NAME	PIN NUMBER	SIGNAL NAME	PIN NUMBER	SIGNAL NAME
J10	MID(28)	K10	MID(20)	L10	NC	M10	NC
J11	MID(26)	K11	MID(23)	L11	MID(18)	M11	NC
J12	MID(27)	K12	MID(24)	L12	V_{DD}	M12	MID(16)
J13	MID(29)	K13	V_{SS}	L13	MID(22)	M13	MID(19)
N1	V_{SS}						
N2	ENREFCLK						
N3	ENRXER						
N4	ENTXD(1)						
N5	NC						
N6	NC						
N7	V_{DDA}						
N8	PMXIN						
N9	ENRXD(3)						
N10	V_{DD}						
N11	NC						
N12	V_{SS}						
N13	MID(17)						

Simplified Block Diagram

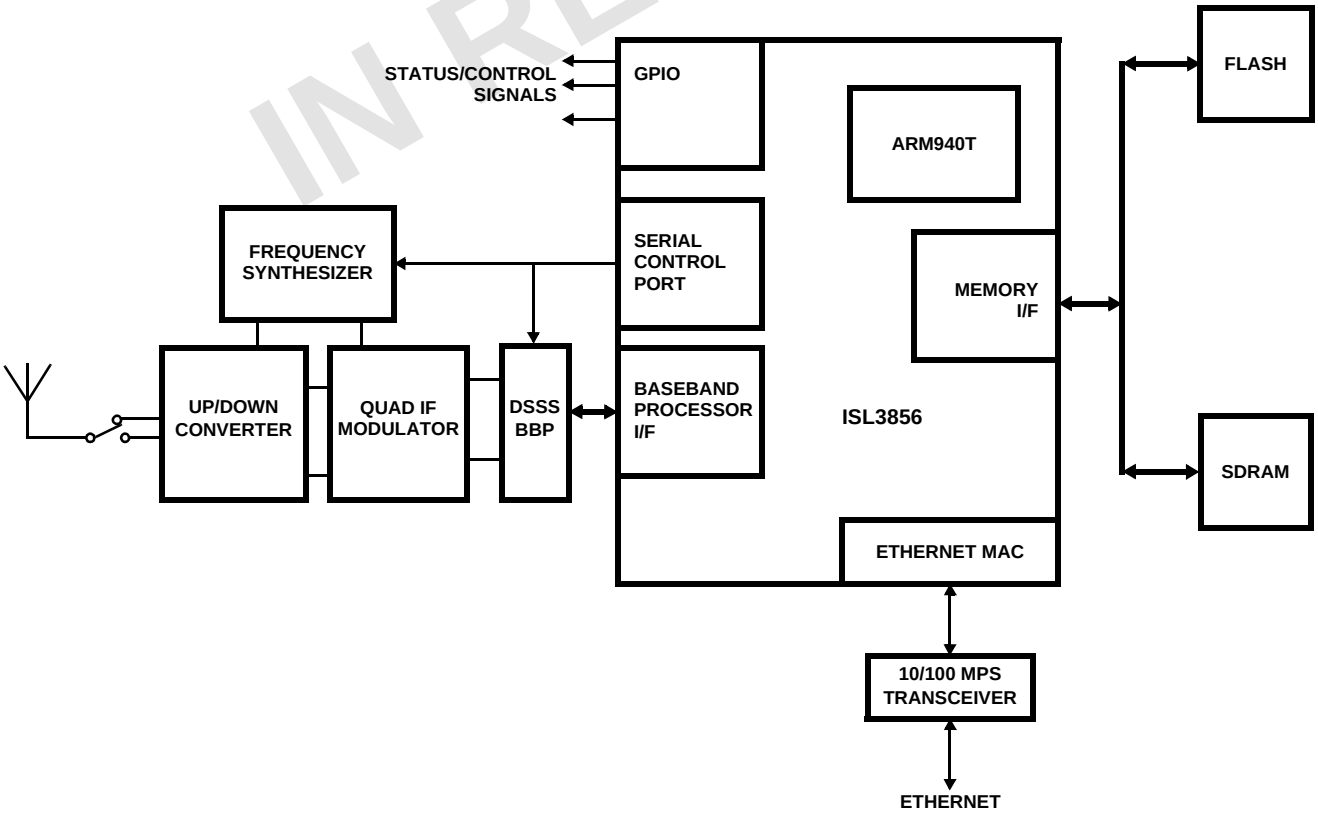


TABLE 1. MEMORY INTERFACE PINS

PIN NAME	BGA PIN NUMBER	PIN I/O TYPE	DESCRIPTION
MIA(17)	A2	CMOS, Output	Memory Address Bit 17
MIA(16)	C3	CMOS, Output	Memory Address Bit 16
MIA(15)	C4	CMOS, Output	Memory Address Bit 15
MIA(14)	B3	CMOS, Output	Memory Address Bit 14
MIA(13)	A3	CMOS, Output	Memory Address Bit 13
MIA(12)	C5	CMOS, Output	Memory Address Bit 12
MIA(11)	B4	CMOS, Output	Memory Address Bit 11
MIA(10)	A4	CMOS, Output	Memory Address Bit 10
MIA(9)	D6	CMOS, Output	Memory Address Bit 9
MIA(8)	E6	CMOS, Output	Memory Address Bit 8
MIA(7)	A5	CMOS, Output	Memory Address Bit 7
MIA(6)	C6	CMOS, Output	Memory Address Bit 6
MIA(5)	D7	CMOS, Output	Memory Address Bit 5
MIA(4)	C7	CMOS, Output	Memory Address Bit 4
MIA(3)	A6	CMOS, Output	Memory Address Bit 3
MIA(2)	B6	CMOS, Output	Memory Address Bit 2
MIA(1)	C8	CMOS, Output	Memory Address Bit 1
MIA(0)	A8	CMOS, Output	Memory Address Bit 0
MID(31)	H12	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 31
MID(30)	H10	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 30
MID(29)	J13	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 29
MID(28)	J10	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 28
MID(27)	J12	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 27
MID(26)	J11	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 26
MID(25)	H9	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 25
MID(24)	K12	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 24
MID(23)	K11	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 23
MID(22)	L13	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 22
MID(21)	J9	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 21
MID(20)	K10	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 20
MID(19)	M13	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 19
MID(18)	L11	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 18
MID(17)	N13	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 17
MID(16)	M12	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 16
MID(15)	D11	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 15
MID(14)	C12	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 14
MID(13)	F8	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 13
MID(12)	E11	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 12
MID(11)	D12	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 11
MID(10)	D13	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 10
MID(9)	F9	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 9
MID(8)	E12	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 8
MID(7)	E13	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 7
MID(6)	F11	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 6
MID(5)	G10	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 5
MID(4)	G11	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 4

TABLE 1. MEMORY INTERFACE PINS (Continued)

PIN NAME	BGA PIN NUMBER	PIN I/O TYPE	DESCRIPTION
MID(3)	G13	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 3
MID(2)	F12	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 2
MID(1)	H11	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 1
MID(0)	H13	CMOS, Bidirectional Data Bus	Memory Bidirectional Data Bus, Bit 0
MICS(3)-	C11	CMOS Output	Memory Chip Select (one per bank). Bit 3
MICS(2)-	B12	CMOS Output	Memory Chip Select (one per bank). Bit 2
MICS(1)-	B13	CMOS Output	Memory Chip Select (one per bank). Bit 1, MICS(1) is typically connected to SADRAM.
MICS(0)-	E10	CMOS Output	Memory Chip Select (one per bank). Bit 0, MICS(0) is typically connected to FLASH memory
MICLK(3)	A11	CMOS Output	Memory Clock Outputs (one per bank) Bit 3
MICLK(2)	B11	CMOS Output	Memory Clock Outputs (one per bank) Bit 2
MICLK(1)	D10	CMOS Output	Memory Clock Outputs (one per bank) Bit 1
MICLK(0)	A12	CMOS Output	Memory Clock Outputs (one per bank) Bit 0
MISA	C10	CMOS Output	Memory Setup Active Output
MIAA	B10	CMOS Output	Memory Access Active Output
MIBLS(3)-	D9	CMOS Output	Memory Byte Lane Strobes (one per byte lane), Bit 3
MIBLS(2)-	B9	CMOS Output	Memory Byte Lane Strobes (one per byte lane), Bit 2
MIBLS(1)-	C9	CMOS Output	Memory Byte Lane Strobes (one per byte lane), Bit 1
MIBLS(0)-	E8	CMOS Output	Memory Byte Lane Strobes (one per byte lane), Bit 0
MIOE-	A9	CMOS Output	Memory Output Enable
MIWE-	B7	CMOS Output	Memory Write Enable
MIRDY	B8	CMOS Input	Memory Ready Input

NOTES:

1. MID[15-8] acts as upper address lines for Flash memory.
2. Signals indicated in table which end with a “-” are active low signals.

TABLE 2. GENERAL PURPOSE I/O PIN ASSIGNMENTS DEFINED FOR PRISM II CONNECTIONS

PIN NAME	BGA PIN NUMBER	PIN I/O TYPE	DESCRIPTION OF FUNCTION (IF OTHER THAN I/O PORT)
GPIO(0)	C1	I/O	PA_PE (Power Amplifier Power Enable Output)
GPIO(1)	D2	I/O	MD_RDY (Input)
GPIO(2)	E5	I/O	RADIO_PD (Radio Power Down Output)
GPIO(3)	F5	I/O	LED_ETHER (Ethernet Activity LED Output)
GPIO(4)	D1	I/O	TX_PE (Transmit Power Enable Output)
GPIO(5)	E3	I/O	MDC (Ethernet Control Interface CLK Output)
GPIO(6)	E4	I/O	MDIO (Ethernet Control Interface I/O)
GPIO(7)	E1	I/O	MAN_RESET (Manual Reset Switch Input)
GPIO(8)	G2	I/O	T/R_SW (Transmit/Receive Switch Output)
GPIO(9)	F3	I/O	T/R_SW# (Transmit/Receive Switch Inverted Output))
GPIO(10)	F1	I/O	MICLKEN1 (Bank 1 Memory Clock Enable Output)
GPIO(11)	F4	I/O	PE1 (Power Enable 1 Output)
GPIO(12)	H2	I/O	PE2 (Power Enable 2 Output)
GPIO(13)	G1	I/O	CAL_EN (Calibration Enable Output)
GPIO(14)	H1	I/O	CCA (Clear Channel Assessment Input)
GPIO(15)	G4	I/O	RESOUT# (Baseband Reset Output), GPIO(15) should be pulled up for regular chip startup operation

TABLE 3. ETHERNET SIGNALS (EN)

PIN NAME	BGA PIN NUMBER	PIN I/O TYPE	DESCRIPTION
ENREFCLK	N2	CMOS Input	MII RXCLK/RMII REFCLK
ENCOL	M9	CMOS Input	MII COL (Collision)
ENRXDV	M8	CMOS Input	MII RX_DV (Receive Data Valid)
ENTXCLK	L6	CMOS Input	MII TX_CLK (Transmit Clock)
ENTXER	K5	CMOS Output	MII TX_ER (Transmit Error)
ENCRSDV	K4	CMOS Input	MII Carrier Sense/RMII CRSDV
ENRXD(3)	N9	CMOS Input	MII Receive Data Input. ENRXD BIT 3, MSB
ENRXD(2)	L8	CMOS Input	MII Receive Data Input. ENRXD BIT 2
ENRXD(1)	J5	CMOS Input	MII Receive Data Input. ENRXD BIT 1
ENRXD(0)	M3	CMOS Input	MII Receive Data Input. ENRXD BIT 0 LSB.
ENRXER	N3	CMOS Input	MII RX_ER (Receive Error Input)
ENTXEN	L4	CMOS Output	MII TX_EN (Transmit Enable Output)
ENTXD(3)	M5	CMOS Output	MII Transmit Data Output. ENTXD BIT 3, MSB
ENTXD(2)	L5	CMOS Output	MII Transmit Data Output. ENTXD BIT 2
ENTXD(1)	N4	CMOS Output	MII Transmit Data Output. ENTXD BIT 1
ENTXD(0)	M4	CMOS Output	MII Transmit Data Output. ENTXD BITR 0, LSB

TABLE 4. RESET AND CLOCK INTERFACE (PM)

PIN NAME	BGA PIN NUMBER	PIN I/O TYPE	DESCRIPTION
PMRES	J8	CMOS Input	System Reset Input
PMXIN	N8	CMOS Input	Oscillator Input (Used as TCLK Input in TIC test mode)
PMXOUT	L7	CMOS Output	Oscillator Output

TABLE 5. TEST INTERFACE (T)

PIN NAME	BGA PIN NUMBER	PIN I/O TYPE	DESCRIPTION
nTRST	H3	CMOS Input	JTAG Test Reset Input (With 50K pull-down resistor)
TMS	J1	CMOS Input	JTAG Test Mode Select Input (With 50K pull-up resistor)
TCLK	J2	CMOS Input	JTAG Test Clock Input (With 50K pull-up resistor)
TDI	H4	CMOS Input	JTAG Test Serial Data Input (With 50K pull-up resistor)
TDO	H5	CMOS Output	JTAG Test Data Serial Output
TST(1)	K2	CMOS Input	Test Mode Select. Sampled on the falling edge of POR. TST(0) is TREQA in TIC test mode.
TST(0)	K1	CMOS Input	Test Mode Select. Sampled on the falling edge of POR. TST(0) is TREQA in TIC test mode.

TABLE 6. SERIAL PORT SIGNALS

PIN NAME	BGA PIN NUMBER	PIN I/O TYPE	DESCRIPTION
SCTXD	C2	Output	Serial Port Transmit Data
SCRXD	D3	Input	Serial Port Transmit Data
SCFRM(2)	B1	Output	Serial Port Frame Synch Output Bit 2
SCFRM(1)	D4	Output	Serial Port Frame Synch Output Bit 1
SCFRM(0)	A1	Output	Serial Port Frame Synch Output Bit 0
SCCLK	B2	Output	Serial Port Clock

TABLE 7. BASEBAND PROCESSOR SIGNALS

PIN NAME	BGA PIN NUMBER	PIN I/O TYPE	DESCRIPTION
BPTXCLK	L2	Output	Baseband Processor Output Transmit Clock Signal
BPTXD	L1	Output	Baseband Processor Output Transmit Data
BPTXRDY	K3	Input	Baseband Processor Input indicating transmit ready
BPTXACT	H6	Output	Baseband Processor Input indicating transmit active should be tied to BPTXRDY
BPRXD	M1	Input	Baseband Processor Receive Data Input
BPRXCLK	M2	Input	Baseband Processor Receive Data Clock
BPRXRDY	L3	Input	Baseband Processor Receive Data Ready Indicator
BPRXACT	J4	Output	Baseband Processor Receive Data Active

TABLE 8. NO CONNECT PINS

PIN NAME	BGA PIN NUMBER	PIN I/O TYPE	DESCRIPTION
NC	H8, K8, K9, L9, L10, M6, M7, M10, M11, N5, N6, N11		No Connect Pins should be tied to V_{DD} with 10K pull-up resistor.

NOTE:

3. All NC pins should be tied to V_{DD} using a 10K pull-up resistor.

TABLE 9. POWER SIGNALS

PIN NAME	BGA PIN NUMBER	PIN I/O TYPE	DESCRIPTION
V_{DD}	A7, A10, A13, D5, F2, F6, F10, G12, J3, J6, L12, N10	Power	Core 3.3V V_{DD} Supply
V_{DDA}	N7	Power	Analog Supply for PLLs
V_{SSA}	K7	Power	Analog GND PLLs
V_{SS}	B5, C13, D8, E2, E7, E9, F7, F13, G3, G5, G6, G7, G8, G9, H7, K6, J7, K13, N1, N12	Power	GND

Absolute Maximum Ratings

Supply Voltage 3.3V
Input, Output or I/O Voltage . . . . . GND -0.5V to V_{DD} +0.5V
ESD Classification . . . . . Class 2

Operating Conditions

Voltage Range . . . . . +3.3V
Temperature Range . . . . . 0^oC to 70^oC

Thermal Information

Thermal Resistance (Typical, Note 4)	θ_{JA} ($^oC/W$)
BGA Package	45

Maximum Storage Temperature Range . . . . . -65^oC to 150^oC
Maximum Junction Temperature . . . . . 100^oC
Maximum Lead Temperature (Soldering 10s) . . . . . 300^oC

CAUTION: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

NOTE:

4. θ_{JA} is measured with the component mounted on a low effective thermal conductivity test board in free air. See Tech Brief TB379 for details.

Electrical Specifications

V_{DD} = 3.0V to 3.3V 10% T_A = 25^oC

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNITS
Power Supply Current	I_{DDOP}	V_{DD} = 3.3V, CLK Frequency 44MHz	-	190 TYP	250 MAX	mA
Input Leakage Current	I_I	V_{DD} = Max, Input = 0V or V_{DD}	-10	1	10	µA
Output Leakage Current	I_O	V_{DD} = Max, Input = 0V or V_{DD}	-10	1	10	µA
Logical One Input Voltage	V_{IH}	V_{DD} = Max, Min	0.7V_{CC}	-	-	V
Logical Zero Input Voltage	V_{IL}	V_{DD} = Min, Max	-	-	V_{CC}/3	V
Logical One Output Voltage	V_{OH}	I_{OH} = -1mA, V_{DD} = Min	V_{CC}-0.2	-	-	V
Logical Zero Output Voltage	V_{OL}	I_{OL} = 2mA, V_{DD} = Min	-	0.2	0.2	V
Input Capacitance	C_{IN}	CLK Frequency 1MHz. All measurements referenced to GND. T_A = 25^oC	-	5	10	pF
Output Capacitance	C_{OUT}	CLK Frequency 1MHz. All measurements referenced to GND. T_A = 25^oC	-	5	10	pF

NOTE:

5. All values in this table have not been measured and are only estimates of the performance at this time.

AC Electrical Specifications

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS
CLOCK SIGNAL TIMING					
OSC Clock Period 16MHz (Typ)	t_{CYC}	56.25	62.5	68.75	ns

ISL3856 System Overview

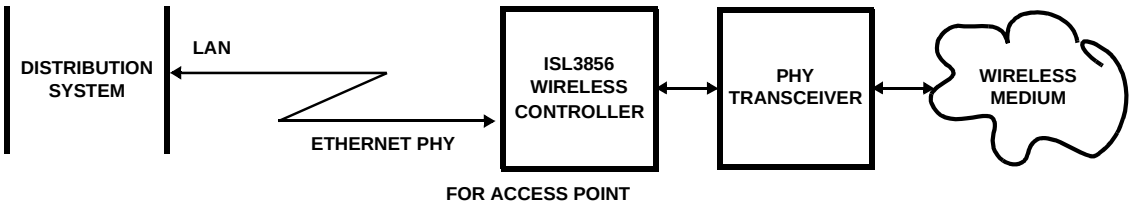


FIGURE 1. TYPICAL APPLICATION

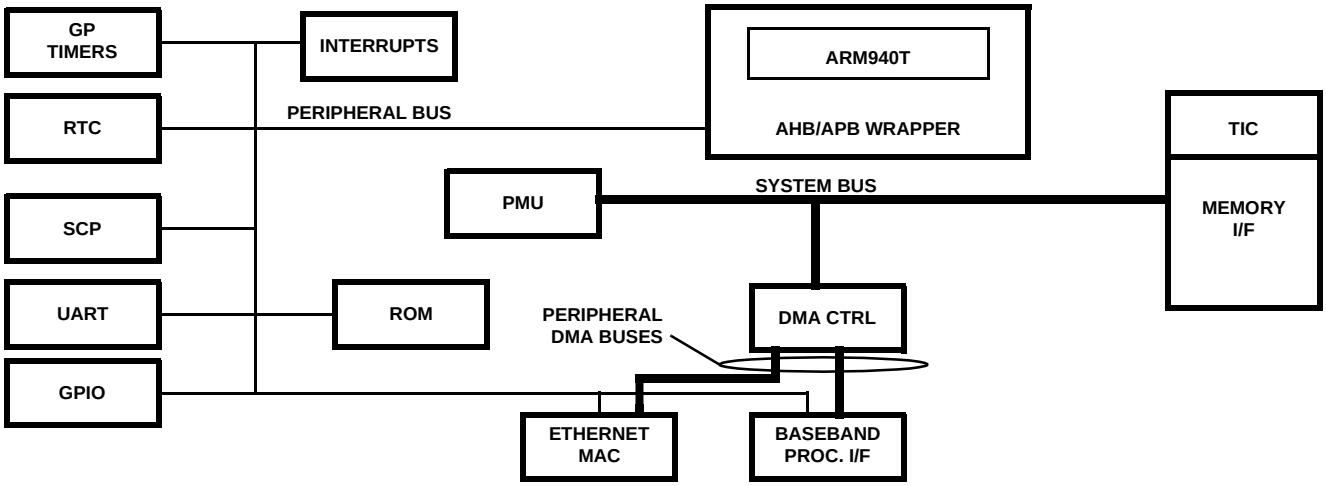


FIGURE 2. ISL3856 BLOCK DIAGRAM

Device Overview

The block diagram for the ISL3856 is shown in Figure 2.

The ISL3856 device is centred around an ARM940T embedded processor. The ARM940T is a high performance processor containing separate data and instruction caches. To provide predictable code behavior, a mechanism for locking code and data into the Instruction and Data caches respectively is provided. This feature can be used to hold high-priority interrupt routines where there is a hard real-time constraint.

The ARM940T processor core used in the ISL3856 device has been equipped with both AHB and APB bus interfaces, by the addition of a wrapper. Using separate system and peripheral buses allows the ARM940T to service interrupts using interrupt service routines locked down in the cache, while the DMA controller transfers data into system memory.

To minimize the pin count a unified memory interface is provided to access all external memory devices. The required flexibility must be incorporated into the memory interface to support FLASH, SRAM and SDRAM devices. The performance of the ISL3856 device is limited by the bandwidth achieved by the memory interface. Therefore, the external memory devices used must be capable of supporting the data bandwidth required.

In order to maximize the data throughput to the external system memory a DMA controller is provided. The DMA controller is used to transfer data between the Ethernet, Baseband Processor interfaces and the system memory.

To allow boot-up and configuration of the device even when the external Flash is not programmed or corrupted, a small on-chip ROM is present. The boot ROM code is executed when the device comes out of reset.

The memory to memory DMA channels of the DMA controller are lower priority than the ARM, to prevent large memory to memory DMA transfers from locking the ARM out from accessing memory, and thereby increasing the worst-case interrupt latency of the CPU.

The clock associated with the APB protocol is specified as PCLK. The clock associated with the AHB protocol is specified as HCLK. Within the ISL3856 device both APB and AHB protocols are used, but are clocked from a single source. To be consistent with the respective bus protocols both PCLK and HCLK signals are referenced in this specification.

ARM940T Processor

ARM940T is a member of the Advanced RISC Machines (ARM) family of general purpose microprocessors. ARM940T is targeted at embedded control applications where high performance and low power are all important. ARM940T supports both the 32-bit ARM and 16-bit THUMB instruction sets allowing the user to trade off between high performance and high code density. ARM940T supports the ARM Debug Architecture and includes logic to assist in both hardware and software debug. ARM940T includes full coprocessor support.

ARM940T is a Harvard architecture cached processor. The separate instruction and data caches in this design are 4kbytes each in size with a four-word line length. A protection unit allows the memory to be segmented and protected in a simple manner, ideal for embedded control applications. There is no virtual to physical address mapping. A write-back cache scheme and write buffer are used to optimize performance and minimize bus traffic thus reducing system power consumption.

Memory Map

The memory map for the ISL3856 is shown in Figure 3. External memory is split into four banks and is accessed through the lower eighth of the address map. The remainder of the memory map is split between the remaining bus slaves.

The on-chip ROM replaces the external memory interface at reset, so that the ARM940T boots from internal memory with the ARM processor at address 0x00000000.

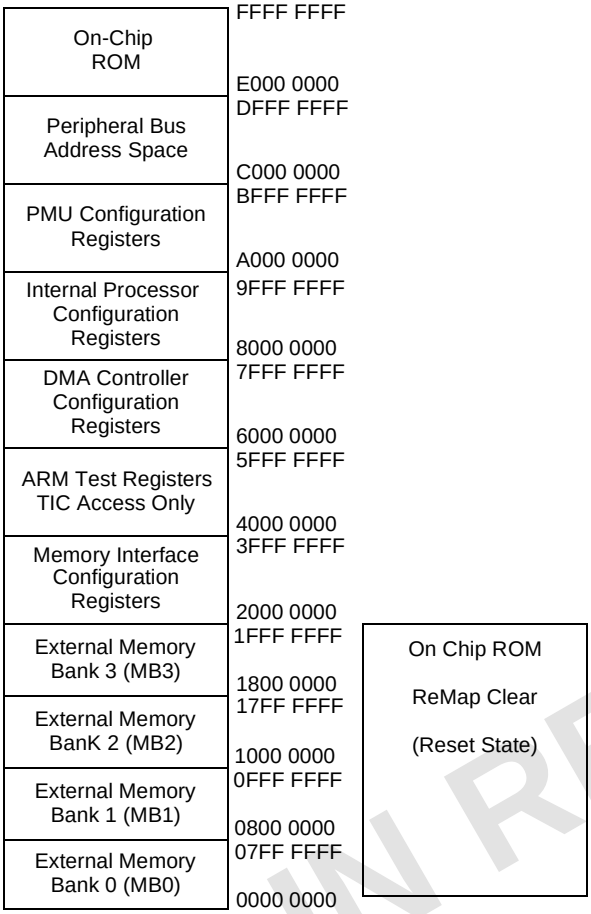


FIGURE 3. ISL3856 MEMORY MAP

On-Chip ROM

The ISL3856 includes 4kbytes of on-chip ROM. ROM normally resides in the top area of the memory map, but also appears at the bottom immediately after reset, to allow the ARM to execute from the ROM when it comes out of reset. Setting the ReMap bit in the PMU removes the ROM from the lower address area. The on-chip ROM resides on the APB bus, allowing on-chip ROM reads to occur simultaneously with AHB DMA cycles.

The ROM size is 4kbytes, organized internally as 1Kx32. Bytes, half-word and word reads are all supported automatically by the ARM, since the ARM correctly selects the addressed byte or halfword from the appropriate byte lane(s).

When coming out of reset, the ARM executes from address 0x00000000, and therefore from the on-chip ROM in it's low address location. Code in the on-chip ROM should jump to code in the ROM at the high address location as soon as possible, and then set the ReMap bit in the PMU to allow access to the external memory. Code must jump to the ROM in the high location before setting the ReMap bit.

If the GPIO[15] bit is sampled LOW at power-on reset, the ReMap bit will automatically be set, forcing the ARM to boot from external Flash. This is primarily a test feature to allow booting if the on-chip ROM contents are corrupt or incorrect. Normally GPIO[15] should be pulled high with a pull-up resistor to ensure that the ARM boots from on-chip ROM.

Memory Interface

Overview

The Memory Interface is used to provide an interface between the system bus and external memory devices. The memory interface supports four independently configurable banks, each capable of supporting burst and non-burst memories, including SDRAMs. Memory components with 32, 16 and 8 bit databuses are supported.

The Memory Interface only supports little endian operation.

To ensure that power is not wasted due to data input buffers floating, the ISL3856 device always re-drives the previous data on the data bus whenever no memory access is in progress. This ensures that the data bus is always driven (except for very short periods of time), and therefore means that external pull-up or pull-down resistors are not needed on the data bus.

Supported Memory Types

The Memory Interface is able to independently configure each of the four memory banks. The memory banks can be configured to support:

- Non-burst read and write access only (e.g., SRAMs).
- Non-burst write accesses, non-burst read accesses and 4 beat burst read accesses (e.g., synchronous FLASH).
- Non-burst write accesses, non-burst read accesses, 4 beat burst read accesses and 4 beat write burst accesses or optionally bursts of undefined length (e.g., synchronous SRAMs).
- SDRAMs accessed by bursts of an undefined length.
- Extended Data Out (EDO) Page Mode memories accessed by bursts of an undefined length.

Burst Access

Burst memory accesses are initiated by outputting the address, asserting the Address Latch Enable output (ALE) and activating the Chip Select (CS). Once the address has been latched by the memory interface, the memory accesses are performed by activating the Write Enable (WE) or the Output Enable (OE) signals for four cycles. Once the burst access is complete all control signals are returned to the inactive state. Each burst access must be four access cycles long and the address must be incremented at the end of each access cycle.

The operation of burst memory devices is assumed to be synchronous to the Memory Clock output i.e., all control signals are latched on the rising edge of the memory clock.

In order to support different speed devices, each memory bank has a dedicated memory clock output. The timing of the burst access is derived from the memory clock for the bank. The 'access cycle' time (i.e., the duration of the actual read or write) is nominally fixed as one memory clock period. The duration of the active ALE pulse (setup cycle) is also nominally fixed as one memory clock period. However, a programmable number of 'decode' cycles can be inserted between the address set-up cycle and the first memory access cycle. Each memory bank can also be configured to insert a minimum number of 'idle' cycles between each burst access.

Non-burst Memory Accesses

Non-burst memory accesses are supported by the memory controller. A non-burst memory access can simply be regarded as a burst access with a burst length of one.

Again the memory access timings are configured relative to the memory clock period and have the same programmability as the burst memory accesses. It may be possible to implement memory accesses without returning to the IDLE state e.g., SRAM read accesses. To support this functionality, each memory bank is able to configure the number of IDLE states to follow a read access and a write access (e.g., an SRAM read may require no IDLE cycles, whereas a write access may require one IDLE cycle).

SDRAM Accesses

SDRAM accesses are performed in the same way to a normal access.

Initially the address is output, the ALE and CS outputs are activated. For SDRAMs the address output during the setup cycle is the row address and the ALE is known as the Row Address Strobe (RAS). During the active cycle the column address is output on the address bus and a Column Address Strobe (CAS) is activated.

The SDRAM data accesses are not performed as a single burst. In each access cycle a new read/write command is issued to the SDRAM device, causing the column address to reload. To support this mode of operation, the SDRAM device must be configured to have a burst length of one. Using a number of individual accesses allows the same location to be accessed in a number of consecutive access cycles e.g., during a byte burst access to a word wide memory bank.

An SDRAM access is not permitted to cross a 512 byte boundary (the minimum supported SDRAM row size) without passing through the SETUP cycle to update the SDRAM's row address. The last access cycle of any transfer must be performed with an auto pre-charge command, or alternatively a precharge command must be issued to the SDRAM. Again the memory access timings are configured relative to the memory clock period and have the same programmability as the normal memory accesses.

A feature of SDRAM read accesses is that there is some latency between the read access being initiated and the data being returned by the device. The data latency from the read command to the first valid data must be configured by the user.

The memory interface is compatible with the Intel "PC SDRAM Specification" and only uses commands defined in that specification.

EDO Memories

EDO memory support is provided at virtually no cost due to its similarity to the SDRAM interface.

To support EDO accesses an ADDRESS WAIT cycle has been added to provide a cycle of setup between the Row address output becoming valid and the RAS signal being activated. Also added is an ACCESS WAIT cycle, after each ACCESS cycle.

The RAS output was used as the Chip Select signal for the SDRAM Interface

EDO read data must be captured by the Memory Interface on the clock cycle following the read access. The single cycle data latency from the read command to the valid data must be configured by the user.

Memory Interface I/O Signals

In order to support FLASH, SRAM and SDRAM devices the I/O pins described in the following sections are provided.

The configuration uses all four banks:

Bank 0 contains 8-bit FLASH

Bank 1 contains 16-bit SRAM

Bank 2 contains 16-bit EDO

Bank 3 contains 32-bit SDRAM.

Chip Selects (MICS[3:0])

Four, active low chips selects are provided, one for each bank. The chip select outputs are used to enable accesses to the devices in the addressed memory bank.

These signals are used to drive the RAS input for EDO memory devices.

Memory Clocks (MICLK[3:0]

Four memory clocks are provided (one for each bank) which are used by synchronous memory mapped devices. All signals generated by the Memory Interface are synchronous to the rising edge of the memory clock.

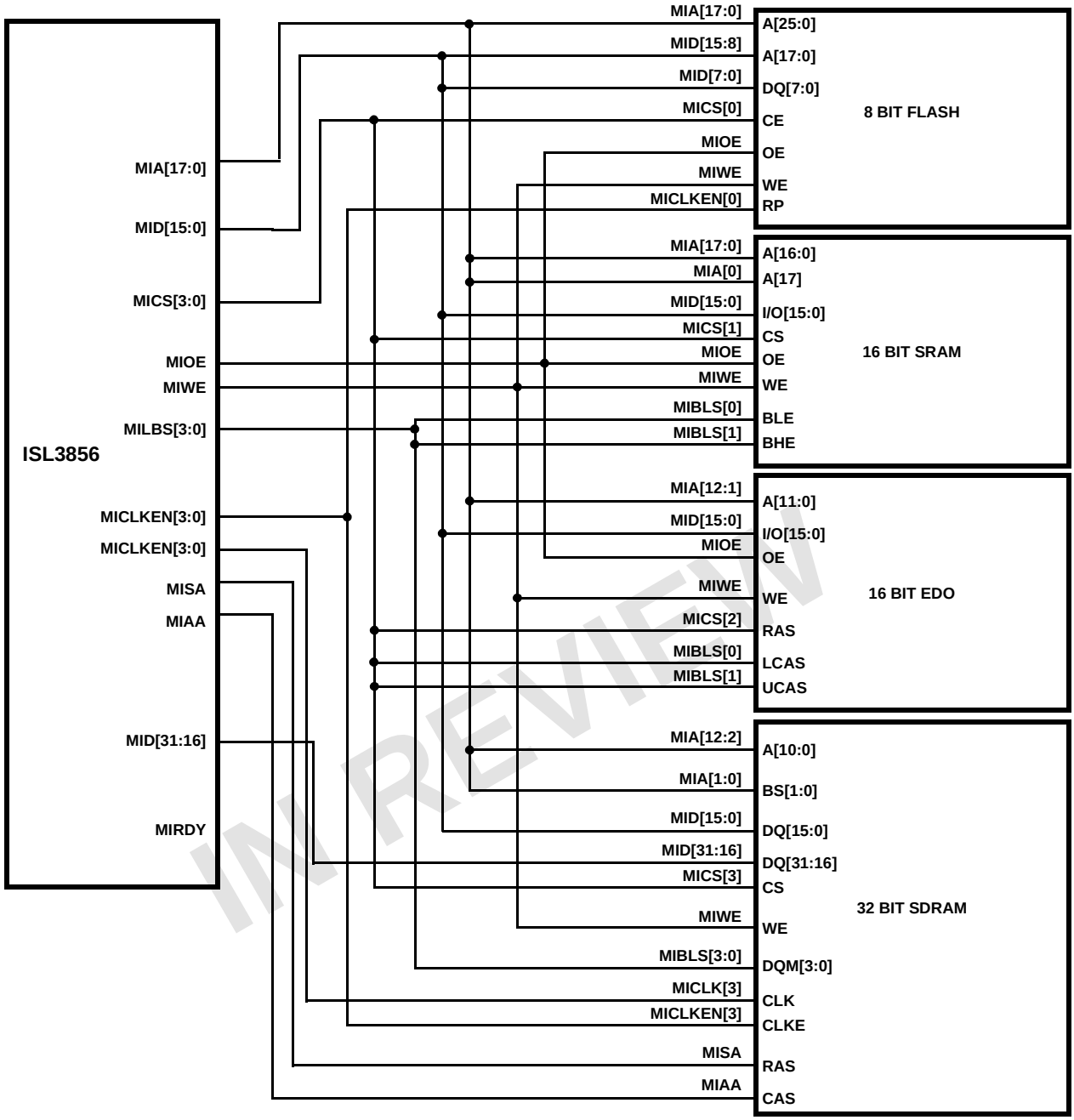


FIGURE 4. ISL3856 MEMORY INTERFACE EXAMPLE

The memory clock output is configurable and is controlled through the memory bank configuration registers. The memory clock output is derived from the system bus clock with a programmable frequency. If the memory clock is not required by the devices in a bank the memory clock output can be disabled. Also, the memory clock output can optionally be disabled when no memory transfer is active.

Memory Clock Enables (MICLKEN[3:0])

Four memory clock enables are provided (one for each bank) which are used to enable and disable the clock within synchronous memory devices. The memory clock enables are multiplexed on GPIO signals to reduce the pin count of the ISL3856 device.

The memory clock enable output can be configured through the memory bank configuration registers to be deactivated when no memory transfer is active.

Setup Active (MISA)

The Setup Active signal is an active low output used to indicate when the Memory Interface is in the SETUP cycle. This signal is used as an Address Strobe for burst FLASH and SRAM devices, and as a Row Address Strobe (RAS) for SDRAM devices.

Access Active (MIAA)

The Access Active signal is an active low output used to indicate when the Memory Interface is in the ACCESS cycle. This signal is used as the Column Address Strobe (CAS) by SDRAM devices.

Output Enable (MIOE)

The Output Enable signal is an active low signal that is active for read operations during the Memory Interface's DECODE, ACCESS and ACCESS WAIT cycles. If a memory data latency is greater then zero the MIOE output will stay active until the last data is read.

Write Enable (MIWE)

The Write Enable signal is an active low signal that is active for write operations. For SDRAM it is active during the Memory Interface's ACCESS and PRECHARGE cycle, for EDO DRAM during DECODE, ACCESS and ACCESS WAIT cycles, for Synchronous SRAM during ACCESS cycles and for SRAM, FLASH and Synchronous FLASH during DECODE cycles.

Byte Lane Strobes (MIBLS[3:0])

The four Byte Lane Strobes are active low outputs used to indicate which bytes of the external memory transfer are valid:

MIBLS[0] - indicates when byte 0 (data[7:0]) is being accessed.

MIBLS[1] - indicates when byte 1 (data[15:8]) is being accessed.

MIBLS[2] - indicates when byte 2 (data[23:16]) is being accessed.

MIBLS[3] - indicates when byte 3 (data[31:24]) is being accessed.

For SDRAM the BLS signals are active during the Memory Interface's ACCESS and PRECHARGE cycles. For EDO DRAM and Synchronous SRAM the signals are active during the ACCESS cycles and for SRAM and FLASH during DECODE cycles.

Data Bus (MID[31:0])

The memory data bus is used to transfer data to and from the external device. The memory devices must be configured so that bit 0 of the data bus is the LSB of the addressed data:

- byte (8 bit) memory banks access data on bits [7:0].
- half word (16-bit) memory banks access data on bits [15:0].
- word (32 bit) memory banks access data on bits [31:0].

Ready (MIRDY)

The memory interface has a ready (RDY) input, which can be used by external devices to lengthen the memory access. When the RDY input is a logic '0' the memory interface is suspended and the system bus is waited (the Memory Clock output continues). When the RDY signal returns to a logic '1' the operation of the memory interface continues.

Address Bus (MIA[17:0])

The 18 bit address bus is used to output the system bus address bits HADD[17:0] in order to provide direct addressing to 256kbytes of external memory space. When supporting byte wide memory devices, the normally unused 8 MSBs (MID[15:8]) of the data bus are used to output the system bus address bits HADD[25:18]. This allows the address range for byte wide memories to be extended to 64Mbytes.

For half-word wide memory devices, the least significant system bus address bit (HADD[0]) is unused. To maximize the accessible address range, bit 0 of the Memory Interface's address output is driven by system bus address bit 18 (HADD[18]). This extends the accessible address space to 512kbytes. For word wide memory devices, the principle is extended by outputting system address bits 18 and 19 (HADD[19:18]) on bits 0 and 1 of the Memory Interface's address output. This extends the accessible address space to 1024kbytes.

In order to further extend the address space, it is possible to output the system bus address bits HADD[26:9] on the Memory Interface address bus during the SETUP cycle. If an external address latch is used to capture the SETUP address, the external address space can be increased to 128Mbytes for each memory bank. The address multiplexing is enabled via in the Memory Bank Configuration Register.

The memory Interface is required to support the SDRAM devices specified in the PC100 SDRAM specification (support 16/32 bit SDRAM devices with 8, 9 or 10 bit column addresses and 8 bit SDRAM devices with 9 or 10 bit column addresses). This requires the column address output to contain at least address bit 11 down to 1 from the system address bus i.e., bits 10 down to 1 for 16 bit memories and bits 11 down to 2 for 32 bit wide memories. In order to provide a continuous memory space, the row address generated must also contain bits 9, 10 and 11 from the system address bus, along with the required number of more significant address bits. To prevent address bits being used in both the row and column addresses to the SDRAM, the column address output (in the ACCESS cycle) must be shifted by one bit position compared to the static memory ACCESS address. This address generation logic is enabled when the memory bank is configured to support SDRAMs.

Ethernet Interface

Overview

The Ethernet Interface is provided to support Ethernet access at both 10Mbits/s and 100Mbits/s data rates. The Ethernet Interface supports full duplex operation by using separate transmit and receive DMA channels.

The device interfaces to an external transceiver through either a Reduced Media Independent Interface (RMII) or a Media Independent Interface (MII). The RMII can be implemented using only the Ethernet Interface I/O signals.

The Serial management pins definitions are implemented as secondary functions in the GPIO.

The user should refer to IEEE802.3 “Local and Metropolitan Area Networks” for complete details on the Ethernet specifications.

Ethernet MAC Macrocell

The Ethernet MAC Macrocells (Rx and Tx) are used to implement data transmission and reception according to the Ethernet Media Access Control protocol. The Ethernet MAC also implements flow control by receiving and sending PAUSE (control) frames.

ISL3856 ETHERNET PIN ASSIGNMENTS

The Ethernet pins for the ISL3856 are described below. Additional information on the functionality of each of the pins can further be found in the IEEE802.3 specification. The user should refer to this document for additional information.

ENREFCLK - The ENREFCLK pin is used to provide the reference clock input to the Ethernet logic for the received data.

ENCOL - The ENCOL pin is used to determine when a collision has taken place. The signal is asserted when a collision is detected on the medium and remains asserted while the collision conditions persist.

ENRXDV - The ENRXDV pin used to indicate that recovered and recorded nibbles on the received data bundle are present and that the data is synchronous with ENREFCLK.

ENTXCLK - The ENTXCLK pin is used as the Ethernet transmit clock. This is a continuous clock that provides the timing reference for ENTXEN, ENTXD and ENTXER signals.

ENTXER - The ENTXER pin indicates an error in transmission of data occurred. This signal is asserted for one or more clock cycles while ENTXEN is also asserted.

ENCRSDV - The ENCRSDV pin indicates when a carrier is detected indicating that either the transmit or receive channel is not idle.

ENRXD (0 - 3) - The receive data pins is a bundle of four data signals ENRXD(0 - 3) that transition with respect to ENREFCLK. ENRXD(0) is the least significant bit.

ENRXER - THE ENRXER pin is asserted for one or more clock cycles to indicate a coding error has been detected. The signal is synchronous with ENREFCLK.

ENTXEN - The ENTXEN pin is used as the Ethernet transmit enable which indicates the device is presenting nibbles on the MII. The pin is asserted by the ISL3856 synchronously with the first nibble of the preamble and remains asserted while all the nibbles are presented to the MII. This signal is negated prior to the first transmit clock following the final nibble of a frame.

Serial Control Port

The serial control port (SCP) is used to provide a half-duplex serial interface to configure and monitor up to three external devices. The SCP functions as a bus master and interfaces to the slave devices using a 4-wire serial protocol. The interface consists of:

Clock (SCCLK) - driven at the bit rate by the SCP.

Frame Sync (SCFRM[2-0]) - driven by the SCP to enable data transfer between the SCP and a slave device.

Transmit Data (SCTXD) - used to transfer data from the SCP to the slave device.

Receive Data (SCRXD) - used to transfer data from the slave device to the SCP.

All interface signals are driven on the falling edge of the clock and all input signals are captured on the rising edge of the clock. When no data transfer is active all interface signals are held in the inactive state, including the clock. To enable up to three slave devices to be driven by the SCP, separate frame sync outputs (**SCFRM[2-0]**) are provided. Only the selected slave device will receive an active frame sync, when a slave device is not selected it must three-state its receive data output.

Additional information on programming and controlling the serial port is contained in the ISL3875 Programmers Manual.

Baud Rate Generation

The baud (or bit) rate is derived by dividing down the peripheral bus clock (PCLK). The clock is divided by a programmable divider, with a range between 1 and 256 as defined in the SCP Control Register. The resultant clock is used to drive the CLK output.

The transmit and receive logic will operate synchronously to the clock output but will actually be clocked directly by the peripheral bus clock.

Interrupt Interface

The interrupt sources for the Serial Control Port are combined to generate a single active high interrupt passed to the system interrupt controller. Each interrupt source can individually be enabled or disabled using the Interrupt Control register. The status of the interrupt sources can be read using the Interrupt Status register. The possible interrupt sources are:

- Rx FIFO full, indicating that 4 words can be read.
- Tx FIFO empty, indicating that 4 words can be written.
- Rx FIFO not empty, indicating that 1 word can be read.
- Tx FIFO not full, indicating that 1 word can be written.

The interrupts are automatically cleared when the FIFOs are serviced.

UART

A single UART is implemented with a 5-pin modem interface.

The UART pins are multiplexed onto the GPIO I/O pins. The UART performs serial-to-parallel conversion on data received from a peripheral device and also, parallel-to-serial conversion on data transmitted to the peripheral device.

Data and control/status information is written and read by the ARM940T via the APB interface. The transmit and receive paths are driven from single byte registers providing a conventional double-buffered UART interface. The UART includes a programmable baud rate generator which generates a common transmit and receive internal clock from the peripheral bus clock (PCLK) input.

The UART offers similar functionality to the industry-standard 16C550 UART device. It supports baud rates of up to 115.2kbits/second. The modem status input signals, CTS, DCD and DSR, are supported. The additional modem status input RI is not supported. Output modem control lines such as RTS and DTR are not explicitly supported. These signals may be implemented using the GPIO lines under software control.

The AMBA UART operation and baud rate values are controlled by the line control register (UARTLCR). The UART can generate four individually maskable interrupts from the receive, transmit and modem status logic blocks. A single combined interrupt is passed to the system interrupt controller which is asserted if any of the individual interrupts are asserted and unmasked.

If a framing, parity or break error occurs during reception, the appropriate error bit is set, and is stored in the receive holding register. If an overrun condition occurs, the overrun register bit is set immediately and the receive holding register's data is prevented from being overwritten.

Baseband Processor Interface

Overview

The baseband processor interface is used to communicate with an off-chip baseband processor using two serial interfaces, one output channel for transmit data and one input channel for receive data. A third bidirectional serial channel for control data transfer is implemented by the Serial Control Port.

Independent Tx and Rx DMA channels are provided to enable the baseband processor interface to access the main memory directly. The interface is configured by the ARM940T via an APB interface.

Data Format Overview

The transmit and receive data to/from the baseband processor has the same format i.e., a header, the data and a CRC checksum. The data is transferred serially, LSB first.

The header format and length can vary for different frame formats. The header length is programmable to enable different frame formats to be supported. The external baseband processor will add/strip the header from the data stream, meaning that a header length of zero bits is valid. The header data length is programmable with a resolution of one bit, but the format will not change while the interface is active.

The data portion of a frame varies in length from frame to frame. The data length can vary from 10 bytes to 2346 bytes, with a resolution of a byte.

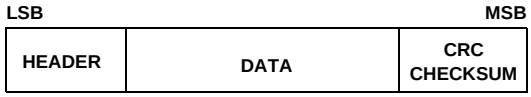


FIGURE 5. BASEBAND PROCESSOR FRAME FORMAT

The transmit CRC checksum is used to detect transmission errors and has a configurable length between 1 and 4 bytes. The checksum generation is performed by the transmitter using the data portion of the frame. The Baseband Processor Interface is able to generate the CRC checksum and append it to the transmit data stream.

Error detection is performed in the receiver by passing the received data stream, including the transmitted CRC checksum, through the same CRC generation logic. If the received frame is error free a known checksum is produced. Errors in the received data can be detected by comparing the generated checksum against the expected value.

The baseband processor interface contains two independent data paths, a transmit data path and a receive data path. Each data path has a separate FIFO used to buffer DMA data.

Transmit Data Serial Protocol

The transmit data is output from the baseband processor interface using the protocol described below:

- The transmit serial interface is activated by asserting the **BPTXACT** output. This powers up the transmit circuitry in the baseband processor.
- When the baseband processor is able to accept data it asserts the **BPTXRDY** signal.
- Data is now serially transferred to the baseband processor, synchronously to the **BPTXCLK** input.
- When the transmit data transfer is complete the **BPTXACT** signal is de-asserted.
- When the baseband processor's transmit circuitry has gracefully disables the TxReady signal is de-asserted. The TxActive signal cannot be re-activated until the TxReady input has been de-asserted.

The Baseband Processor Interface is only required to drive the **BPTXACT** and **BPTXD** outputs, all the other Tx signals, **BPTXRDY** and **BPTXCLK** are inputs.

The polarity of all the control signals (**BPTXACT**, **BPTXRDY** and **BPTXCLK**) in the transmit data serial protocol are programmable via the Transmit Control Register.

The **BPTXRDY** input should be directly connected to the **BPTXACT** output.

Receive Data Serial Protocol

The receive data protocol is similar to the transmit data protocol:

- The receive interface is enabled by asserting the **BPRXACT** output. This powers up the receive circuitry in the baseband processor.
- When the baseband processor has received data it asserts the **BPRXRDY** signal.

- Data is now serially transferred from the baseband processor, synchronously to the **BPRXCLK** input.
- When the receive data transfer is complete the **BPRXACT** signal is de-asserted.
- When the baseband processor's receive circuitry has gracefully disabled, the BPRXRDY signal is de-asserted. The **BPRXACT** signal cannot be re-activated until the **BPRXRDY** input has been de-asserted.

The Baseband Processor Interface is only required to drive the **BPRXACT** output, all the other Rx signals are inputs.

The polarity of all the control signals (**BPRXACT**, **BPRXRDY** and **BPRXCLK**) in the receive data serial protocol are programmable via the Receive Control Register.

If the **BPRXACT** output is not required by the external baseband processor the signal can be left unconnected. If the **BPRXRDY** input is not supported by the baseband processor, it can be directly connected to the **BPRXACT** output.

GPIO Interface

Overview

The GPIO provides up to 16 lines of general purpose I/O. Each GPIO line is completely general purpose, and may be used as an output, an open-drain output, an input, or an interrupt input.

Many of the GPIO lines can be used to provide alternative functionality required for different system configurations.

Functional Description

The GPIO port has data and data direction registers. The data register is used to set the values of pins that are configured as outputs. The data direction register is used to configure each GPIO line as either an input or an output. The status register is used to read the value of the GPIO pins, both the inputs and outputs.

Each GPIO input can be configured to generate an interrupt from an active high level, an active low level, a rising edge or a falling edge. The GPIO interrupts are combined to generate two interrupt signals passed to the interrupt controller. Each of the interrupts has an independent enable register, allowing both FIQ and IRQ interrupts to be selectively generated from the GPIO pins.

The GPIO can easily be used to implement an open drain output. The data register bit for the required line should be written with a zero, the data direction register is then used to switch between input and output modes. If the GPIO line is configured as an output the pin is driven low, if it is configured as an input, the output is three-stated. The GPIO line can of course be used to generate interrupts (either falling edge or active low) in the normal way, to generate an interrupt if an external device drives the signal low.

The functionality of each of the GPIO lines is selected by the Function Select register. This enables either the normal GPIO functionality or a secondary function.

When the secondary operation is enabled, the GPIO pin's data and data direction control signals are ignored and the interrupt generation logic is disabled. When the GPIO operation is selected, the secondary input signals are forced to a logic '0' and output signals are ignored.

Complete details on programming the GPIO pins can be found in the ISL3856 Programmers Manual.

Interrupt Controller

Overview

The interrupt controller provides a simple software interface to the interrupt system. Two types of interrupt are available in ARM systems:

- FIQ (Fast Interrupt Request) for fast, low latency interrupt handling
- IRQ (Interrupt Request) for more general interrupts

Separate interrupt controllers are used for FIQ and IRQ. These interrupt controllers use a bit location for each different interrupt source. No hardware priority scheme or any form of interrupt vectoring is provided, because these functions can be provided in software.

All interrupt source inputs to the interrupt controller block are active HIGH and level sensitive. Any inversion or latching required to provide edge sensitivity is provided in the block generating the source of the interrupt.

A programmed interrupt register is also provided to generate an interrupt under software control. Typically, this may be used to downgrade a FIQ interrupt to an IRQ interrupt, where performing all the operations required in the FIQ handler would compromise the FIQ latency.

The interrupt controller provides interrupt source status, interrupt request status and an enable register. The enable register is used to determine whether or not an active interrupt source should generate an interrupt request to the processor.

The interrupt source status indicates whether or not the appropriate interrupt source is active prior to masking and the interrupt request status indicates whether or not the interrupt source is causing a processor interrupt.

The enable register has a dual mechanism for setting and clearing the enable bits. This allows enable bits to be set or cleared independently, with no knowledge of the other bits in the enable register. When writing to the enable set location, each data bit that is HIGH sets the corresponding bit in the enable register. All other bits of the enable register are unaffected. Conversely, the enable clear location is used to clear bits in the enable register while leaving other bits unaffected.

For complete details on programming and controlling interrupts, refer to the ISL3856 Programmers Manual.

Power Management Unit

The Power Management Unit (PMU) is used to generate the clocks and reset signals for the ISL3856 device. The clocks are generated by an on-chip oscillator (using an external oscillator) and a Phase Locked Loop (PLL).

Functional Description

The PMU will generate two main clocks; a core clock and a system clock:

- Core Clock - the core clock is used to clock the ARM940T core and has a frequency of twice the system clock. When in FastBus mode the ARM940T is clocked by the system clock and the core clock is disabled. The Core Clock has a maximum frequency of 160MHz.
- System Clock - the system clock is used by all internal buses and peripherals. The system clock source can be selected from the 16MHz oscillator clock or the output of the PLL divided by two. The System Clock has a maximum frequency of 80MHz.

The default selection for the system clock after reset is the output of the 1MHz clock derived from the on-chip oscillator. After reset the core clock is disabled and the ARM940T is in FastBus mode.

Additionally, the PMU can generate an external clock, which may be optionally output on one of the GPIO lines. This clock may be used to clock an external radio chip set.

In order to implement reliable clock switching a status bit is generated which indicates when the PLL has had sufficient setup time to stabilize the outputs. Also, all clock switching is performed using circuitry designed to produce glitch free outputs.

The PMU also generates a 1MHz clock-enable signal synchronous to the system clock. This signal can be used for timing purposes within the ISL3856 device. The purpose of the clock enable is to ensure that the timing functions within the ISL3856 device are performed at the correct rate regardless of the system clock frequency produced by the PMU.

The PMU is able to suspend the operation of the ARM940T when the CoreStop bit is asserted in the PMU Control Register. This mode is automatically cleared when an interrupt (IRQ or FIQ) is activated by the interrupt controller.

The external clock (ECLK) is generated by dividing the core clock by a programmable value. ECLK can be output on one of the GPIO lines.

Clock Generation

The system and core clocks are generated by a PLL driven from the 16MHz on-chip oscillator.

Power Management and System Bus Arbitration

To control the device's power consumption the PMU is able to change the arbitration priority of the system bus. The following modes are supported:

Processor Suspend: when the Processor Suspend bit is set in a write to the System Bus Mode Control Register, the PMU prevents the completion of the ARM940T's write cycle by granting control of the system bus to another master. As the processor's internal clocks are disabled until a write is completed, this will suspend the processor's operation. This mode is cleared by a FIQ or optionally by an IRQ from the interrupt controller i.e., the ARM940T is granted mastership of the system bus to complete the write cycle.

System Bus Suspend: when the System Bus Suspend bit is set in a write to the System Bus Mode Control Register, a write to the System Suspend register will cause the PMU to hold the system bus. This will suspend the ARM940T and prevent any accesses on the system bus. When entering this mode, it is possible to disable the clocks to the Memory Interface and DMA controller through the System Bus Clock Stop bit. This mode is cleared by a FIQ or optionally by an IRQ from the interrupt controller i.e., the write is allowed to complete and normal system bus operation will resume.

To improve interrupt latency it is possible to automatically increase the priority of the ARM940T on the system bus following an interrupt. When this function is enabled, any FIQ and optionally any IRQ will cause the ARM940T to be become top priority on the system bus. The system bus priorities are returned to normal through the System Bus Mode Status Register.

Reset Generation

The ISL3856 device is reset from two sources, an internally generated power on reset (POR) and an external active high system reset input (**PMRES**). The ARM940T processor core can also be reset from the Real Time Clock (using the STBARK signal) or from software, by writing to the PMU Status register.

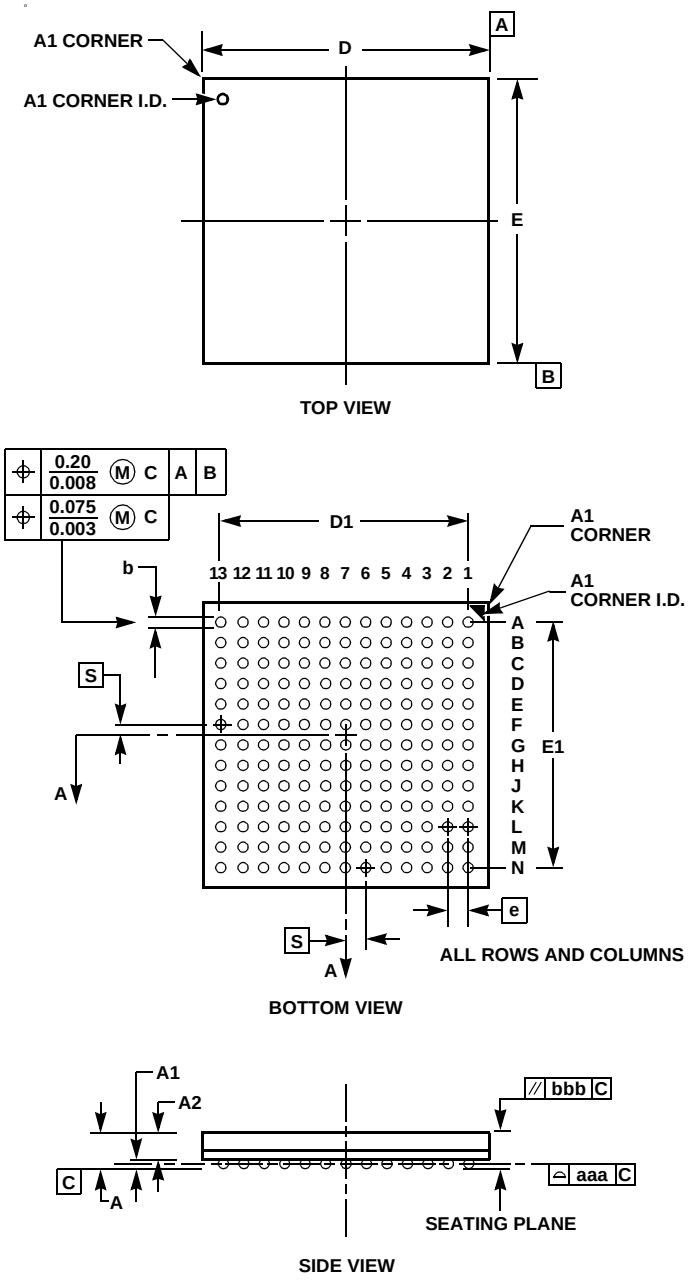
The source of the last reset generated can be read from the PMU Status Register.

JTAG Test Interface

The ISL3856 device contains a JTAG Interface which allows in-circuit debugging. The whole interface is controlled via 5 dedicated pins: TDI, TMS, TCK, nTRST and TDO.

Further details on the JTAG interface and software support are provided in the programmer's manual.

Plastic Ball Grid Array Packages (BGA)



V169.11x11

169 BALL PLASTIC BALL GRID ARRAY PACKAGE

SYMBOL	INCHES		MILLIMETERS		NOTES
	MIN	MAX	MIN	MAX	
A	-	0.057	-	1.45	-
A1	0.008	0.014	0.20	0.35	-
A2	0.039	0.043	1.00	1.10	-
b	0.016	0.021	0.40	0.55	7
D/E	0.425	0.441	10.80	11.20	-
D1/E1	0.370	0.386	9.40	9.80	-
N	169		169		-
e	0.032 BSC		0.80 BSC		-
MD/ME	13 x 13		13 x 13		3
bbb	-	0.008	-	0.20	-
aaa	-	0.006	-	0.15	-

Rev. 0 9/00

NOTES:

1. Controlling dimension: MILLIMETER. Converted inch dimensions are not necessarily exact.
2. Dimensioning and tolerancing conform to ASME Y14.5M-1994.
3. “MD” and “ME” are the maximum ball matrix size for the “D” and “E” dimensions, respectively.
4. “N” is the maximum number of balls for the specific array size.
5. Primary datum C and seating plane are defined by the spherical crowns of the contact balls.
6. Dimension “A” includes standoff height “A1”, package body thickness and lid or cap height “A2”.
7. Dimension “b” is measured at the maximum ball diameter, parallel to the primary datum C.
8. Pin “A1” is marked on the top and bottom sides adjacent to A1.
9. “S” is measured with respect to datum’s A and B and defines the position of the solder balls nearest to package centerlines. When there is an even number of balls in the outer row the value is “S” = e/2.

Sales Office Headquarters

NORTH AMERICA

Intersil Corporation
7585 Irvine Center Drive
Suite 100
Irvine, CA 92618
TEL: (949) 341-7000
FAX: (949) 341-7123

Intersil Corporation
2401 Palm Bay Rd.
Palm Bay, FL 32905
TEL: (321) 724-7000
FAX: (321) 724-7946

EUROPE

Intersil Europe Sarl
Ave. William Graisse, 3
1006 Lausanne
Switzerland
TEL: +41 21 6140560
FAX: +41 21 6140579

ASIA

Intersil Corporation
Unit 1804 18/F Guangdong Water Building
83 Austin Road
TST, Kowloon Hong Kong
TEL: +852 2723 6339
FAX: +852 2730 1433