CX805 – 20 Advance Information

Baseband Processor for Multiband GSM and GPRS Applications

The Skyworks CX805-20 Baseband Processor (BP) is a highly integrated, dual core processor optimized for use in Global System for Mobile communications™ (GSM™) and General Packet Radio Service (GPRS) cellular handset applications. The CX805-20 is the baseband portion of the Skyworks GSM/GPRS System Solution.

Both the Digital Signal Processor (DSP) core and the ARM7 THUMB™ Reduced Instruction Set Computing (RISC) architecture are well suited to meet the needs of low power, high performance embedded systems such as cellular phones. The BP operates over a range of 2.7 V to 3.3 V.

The baseband processing tasks are divided between the DSP and ARM7 processor cores. The DSP core executes the physical layer, layer 1, processing functions, and the ARM microcontroller core executes Layer 2 and Layer 3 protocol software and the Man-Machine Interface (MMI) functions. The two cores communicate through a dedicated block of dual port memory. Each of the functional blocks in the device can be individually powered down to ensure minimum current consumption in the idle or standby mode.

The ARM and DSP peripherals are shown in Table 1. The CX805-20 block diagram is provided in Figure 1.

The BP is packaged in a 12 x 12 mm, 160-pin Fine Pitch Ball Grid Array (FPBGA) with a 16-bit data bus and a 24-bit address bus. The BP package and pin configurations are shown in Figure 2. The signal pin assignments and functional pin descriptions are provided in Table 2.

Features

- 0.25 μm CMOS process technology
- ARM7 TDMI core
- Skyworks DSP core with all memory on-chip
- GPRS class 10 and circuit-switched data (14.4 kbps) data services
- Half-rate, full-rate, and enhanced full-rate speech coders
- Voice features such as voice recognition, conversation record, and voice memo
- Hardware accelerator for GPRS encryption algorithms (GEA 1 and 2)
- Integrated Real Time Clock (RTC)
- Interface to handset MMI peripherals such as keypad, Liquid Crystal Diode (LCD), annunciator
- Interfaces to Skyworks IA and PMIC devices
- Interface to Subscriber Identity Module (SIM)
- Addresses up to 4 MB of external memory (FLASH or SRAM)
- Application Interfaces:
  - Serial/RS-232
  - Infra-red Data Adapter (IrDA)
- Low power operation. 3.0 V I/Os and an on-chip supplied 2.5 V core
- Five Chip Select (CS) signals for external memory
- 160-pin FPBGA 12x12 mm package

Applications

- GSM handsets and modules (900/1800/1900 MHz)
- GPRS handsets and modules (900/1800/1900 MHz)
- Bluetooth-enabled wireless headset modules

Table 1. ARM and DSP Peripherals

ARM Peripherals		DSP Peripherals
GPRS Engine	Keypad Interface block	Cidex Engine
Six Direct Memory Access (DMA) channels	Autobaud block supporting one of the serial ports	DMA Port connected to a Serial Port
Two General Purpose Timers (GPTs)	Clock Control block, which also provides interface logic between cores	Four serial ports interfacing off-chip (one supporting high-speed debug)
Gendex Engine	RTC	Viterbi Engine
Calibration and Sleep Timer	Escape Detection block	General Purpose Timer
Two Universal Asynchronous Receive/Transmits (UARTs) with IrDA capability	In-house Serial Port similar to Inter-Integrated Circuit (I2C)	Control block which supports some interface logic to the ARM and a test mux control block
PWM (Tone Generator)	Test Control block	Trap and Patch Support Logic
Cyclic Redundancy Check (CRC) Generator	SIM Interface (I/F) block	Interrupt Controller (IC)
27 bits of General Purpose Input/Output (GPIO)	Interrupt Controller (IC)	

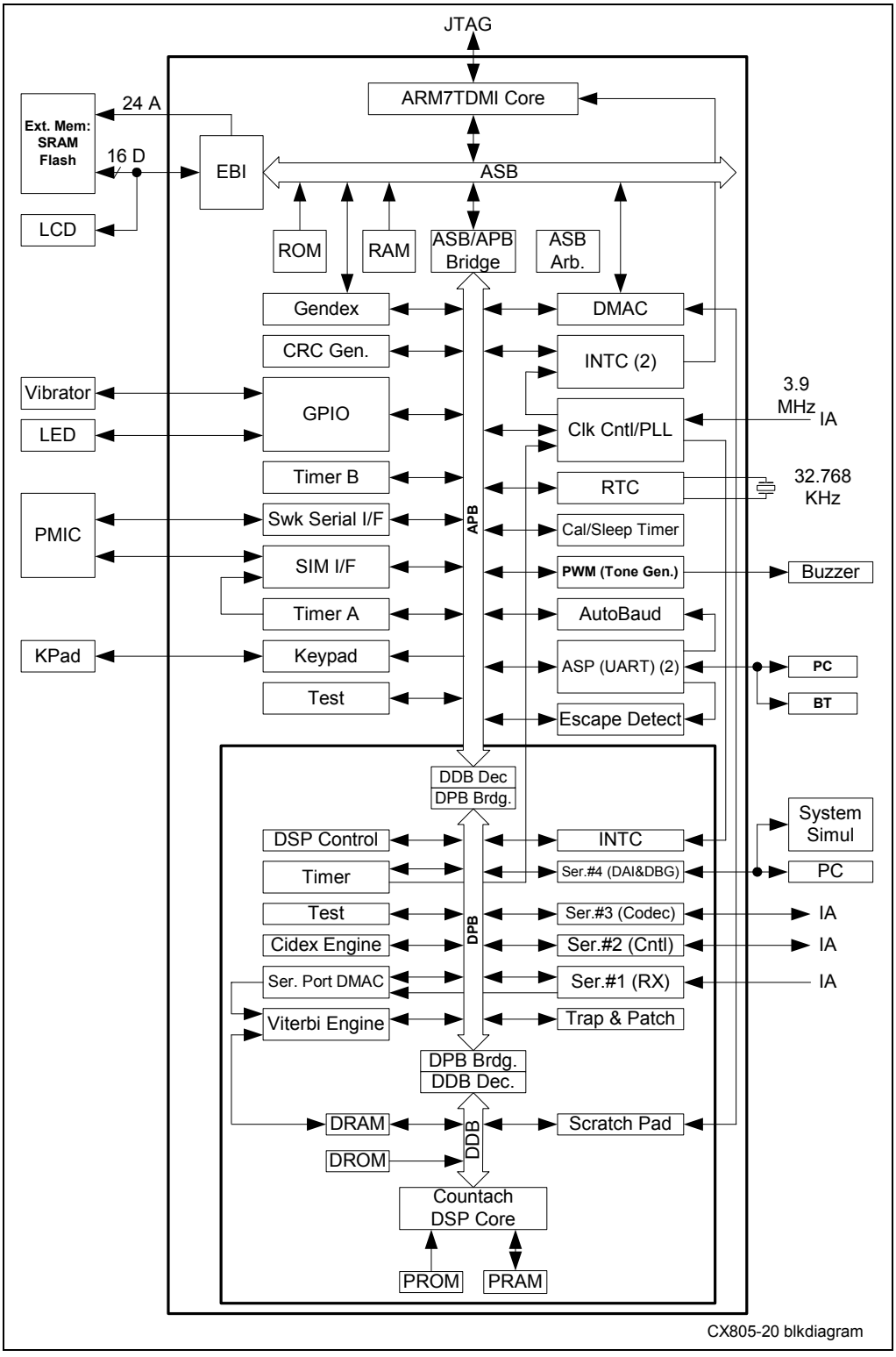


Figure 1. CX805-20 Block Diagram

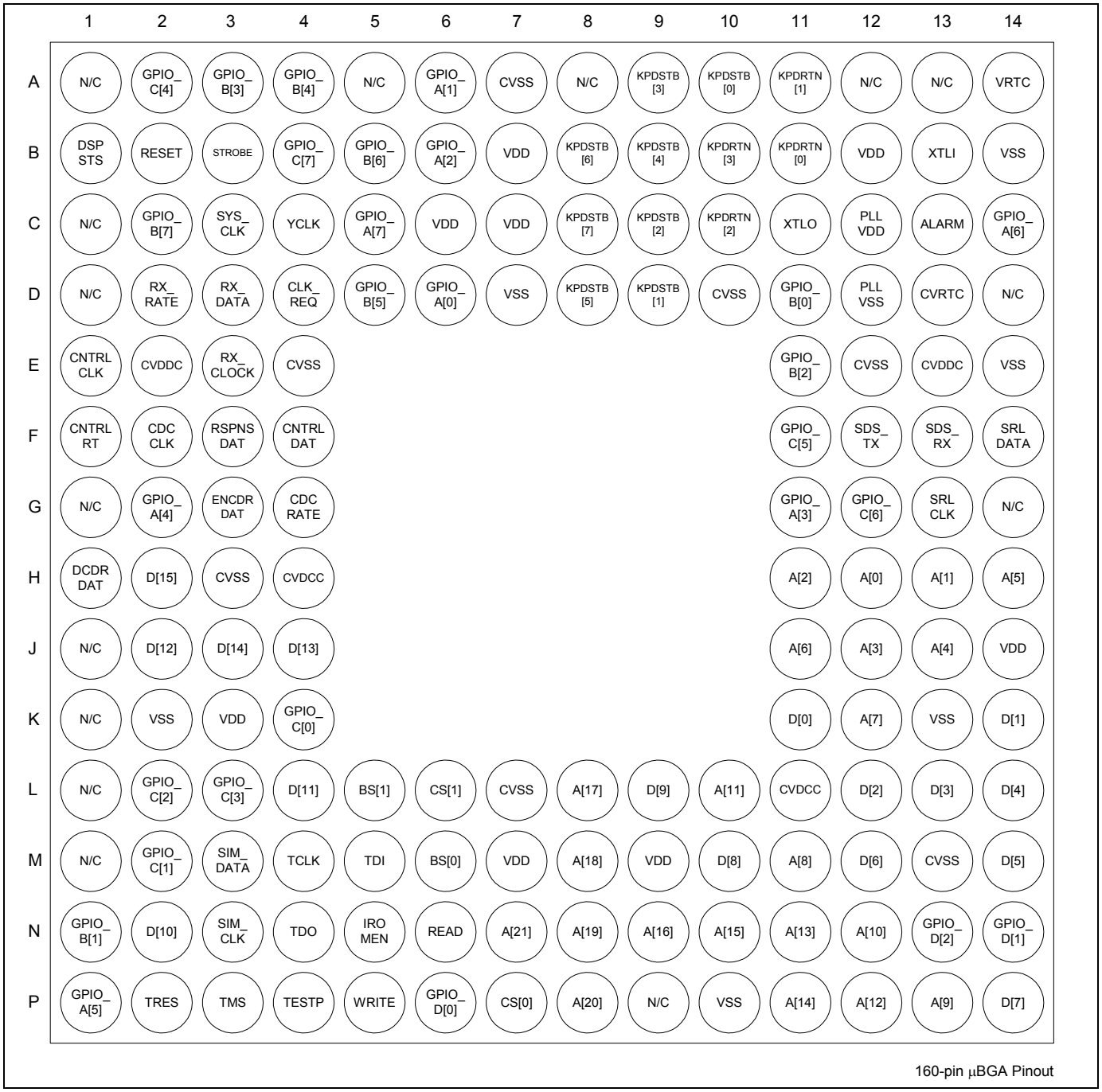


Figure 2. BP Device 160-Pin FPBGA Pinout (Top View)

Table 2. BP Pinout Assignments (1 of 2)

Pin #	Pin Name	Description	Pin#	Pin Name	Description
Power Supply					
E2, E13, H4, L11	CVDDC	Supply Out Core	C12	PLLVDD	PLL supply
A7, D10, E4, E12, H3, L7, M13	CVSS	Core ground	D12	PLLVSS	PLL ground
B7, B12, C6, C7, J14, K3, M7, M9	VDD	I/O pins supply	A14	VRTC	Input/Output Ring for RTCsuplly
K2, P10, K13, E14, B14, D7	IOVSS	I/O pins ground	D13	VRTC	RTC – Supply Inl
Test/JTAG					
M5	TDI	JTAG data In	M4	TCLK	JTAG clock
N4	TDO	JTAG data Out	P2	TRES	JTAG reset
P3	TMS	JTAG mode select	N5	IROMEN	Enable Internal ROM
System					
B2	RESET	Power-on reset	B3	STROBE	Test
D4	CLK_REQ	Clock request signal	C13	ALARM	RTC alarm
C3	SYS_CLK	System clock	B13	XTLI	32 kHz crystal input
C4	YCLK	Test	C11	XTLO	32 kHz crystal output
B1	DSPTST	Test	P4	TESTP	Test
External Bus					
H12	A[0]	Address bus 0 to 21	K11	D[0]	Data bus 0 to 15
H13	A[1]		K14	D[1]	
H11	A[2]		L12	D[2]	
J12	A[3]		L13	D[3]	
J13	A[4]		L14	D[4]	
H14	A[5]		M14	D[5]	
J11	A[6]		M12	D[6]	
K12	A[7]		P14	D[7]	
M11	A[8]		M10	D[8]	
P13	A[9]		L9	D[9]	
N12	A[10]		N2	D[10]	
L10	A[11]		L4	D[11]	
P12	A[12]		J2	D[12]	
N11	A[13]		J4	D[13]	
P11	A[14]		J3	D[14]	
N10	A[15]		H2	D[15]	
N9	A[16]		N6	READ	Read strobe
L8	A[17]		P5	WRITE	Write strobe
M8	A[18]		P7	CS[0]	Flash select, chip select 0
N8	A[19]		L6	CS[1]	RAM select, chip select 1
P8	A[20]		M6	BS[0]	Byte select for 16-bit SRAM (lower byte)
N7	A[21]		L5	BS[1]	Byte select for 16-bit SRAM (upper byte)

Table 2. BP Pinout Assignments (2 of 2)

Pin #	Pin Name	Description	Pin#	Pin Name	Description
Keypad Control					
A10	KPDSTB[0]	Keypad strobe lines 0 to 7	B8	KPDSTB[6]	Keypad strobe lines 0 to 7
D9	KPDSTB[1]		C8	KPDSTB[7]	
C9	KPDSTB[2]		B11	KPDRTN[0]	Keypad return lines 0 to 3
A9	KPDSTB[3]		A11	KPDRTN[1]	
B9	KPDSTB[4]		C10	KPDRTN[2]	
D8	KPDSTB[5]		B10	KPDRTN[3]	
GPIO					
D6	GPIO_A[0]	GPIO signal port A[0]	K4	GPIO_C[0]	GPIO port C[0] (I/O select bit = 0)
A6	GPIO_A[1]	GPIO signal port A[1] (I/O select bit = 0) Keyboard return 4 (I/O select bit = 1)	M2	GPIO_C[1]	GPIO port C[1] (I/O select bit = 0) SIM_ENABLE (I/O select bit = 1)
B6	GPIO_A[2]	GPIO signal port A[2] to A[7]	L2	GPIO_C[2]	GPIO port C[2] (I/O select bit = 0)
G11	GPIO_A[3]		L3	GPIO_C[3]	GPIO port C[3] (I/O select bit = 0) SIM_RESET (I/O select bit = 1)
G2	GPIO_A[4]		A2	GPIO_C[4]	GPIO signal port C[4] (I/O select bit = 0)
P1	GPIO_A[5]		F11	GPIO_C[5]	GPIO signal port C[5] (I/O select bit = 0) DEBUG_TX (I/O select bit = 1)
C14	GPIO_A[6]		G12	GPIO_C[6]	GPIO signal port C[6] (I/O select bit = 0) DEBUG_RX (I/O select bit = 1)
C5	GPIO_A[7]		B4	GPIO_C[7]	GPIO signal port C[7]
D11	GPIO_B[0]	GPIO signal port B[0] to B[7]	P6	GPIO_D[0]	GPIO signal port D[0] (I/O select bit = 0) CS[2] (I/O select bit = 1)
N1	GPIO_B[1]		N14	GPIO_D[1]	GPIO signal port D [1] (I/O select bit = 0) CS[3] (I/O select bit = 1)
E11	GPIO_B[2]		N13	GPIO_D[2]	GPIO signal port D[2] (I/O select bit = 0) CS[4] (I/O select bit = 1)
A3	GPIO_B[3]				
A4	GPIO_B[4]				
D5	GPIO_B[5]				
B5	GPIO_B[6]				
C2	GPIO_B[7]				
Serial Data Port					
F13	SDS_RX	SDS port data in	F14	SRLDATA	Skyworks serial bus data I/O
F12	SDS_TX	SDS port data out	G13	SRLCLK	Skyworks serial bus clock
IA Serial Ports					
D3	RX_DATA	Receive port data	F1	CNTRLRT	Control port rate
E3	RX_CLOCK	Receive port clock	H1	DCDRDAT	Codec port data to IA
D2	RX_RATE	Receive port rate	G3	ENCDRDAT	Codec port data from IA
F4	CNTRLDAT	Control port data to IA	F2	CDCCLK	Codec port clock
F3	RSPNSDAT	Control port data from IA	G4	CDCRATE	Codec port rate
E1	CNTRLCLK	Control port clock			
SIM					
N3	SIM_CLK	Clock signal for SIM interface	M3	SIM_DATA	Bi-directional SIM data signal

Technical Description

The BP is a dual-core device consisting of an ARM7 THUMB microcontroller core, a Skyworks proprietary DSP core, and all the digital control circuitry required in a GSM handset. The following sections describe the operation and programming of each of the BP functional blocks. Table 5 specifies the address and default value for each of the registers in the device.

Note. *Table 5 specifies the value of each register before the BP IROM code is executed.*

ARM7 TDMI Core

The ARM7 THUMB core is a member of the ARM family of general-purpose 32-bit microcontrollers that offer high performance with very low power consumption. The ARM architecture is based on RISC principles with a simple yet powerful instruction set. This simplicity enables high instruction throughput and rapid real-time interrupt response. This core has the following features:

- 32/16-bit RISC architecture
- ARM 32-bit instruction set for maximum performance and flexibility
- Thumb 16-bit instruction set for increased code density
- Unified 32-bit data bus carries both instructions and data
- 32-bit ALU and high performance multiplier
- Fully static operation
- Coprocessor interface
- Extensive debug facilities

Note. *For further information on ARM7 THUMB, please refer to the ARM7TDMI data sheet published by ARM.*

Internal Memory

Twelve kB of Internal Random Access Memory (IRAM) and 16 kB of Internal Read Only Memory (IROM) support the BP. Both IRAM and IROM directly interface to the 32-bit data and address buses from the ARM microcontroller core.

The IROM contains the embedded firmware, which is executed on power up. The IRAM is used for data storage during run time.

Both internal memory blocks are single cycle access.

System Integration Unit

The System Integration Unit (SIU) is used to interface the 32-bit ARM bus to 8-, 16-, or 32-bit memory and peripherals connected to the External Bus Interface (EBI) and ARM Peripheral Bus (APB). Since the ARM only interfaces to 32-bit devices, the SIU formats the address and data to and from the ARM to allow 8-, 16-, or 32-bit data transfers.

The SIU also performs address decoding to generate internal and external CS signals. These peripherals can be:

- Peripherals internal to the BP, for example, the Pulse Width Modulator (PWM), which interface to the APB.
- External peripherals, for example, system Flash memory, which interface to the EBI.

The SIU performs the following main functions:

- Generates the required internal or external chip selects.
- Formats the address bus and data bus for 8-, 16-, or 32-bit transfers.

Separating the buses minimizes power dissipation, since only the required bus lines are driven at any one time.

ARM Peripheral Bus

The APB interfaces to internal peripherals on the BP. The bus supports both 8-bit and 16-bit peripherals. The bus is only active when one of the internal peripherals is being accessed. If a device on the EBI is being accessed, there is no activity on the APB.

External Bus Interface

The EBI allows external memory devices, for example, Flash, Static Random Access Memory (SRAM), to be connected to the BP. Internal to the BP, the EBI is connected to the SIU. The device features a 16-bit data bus D[15:0] and a 22-bit address bus A[21:0]. EBI supports the following features:

- Support asynchronous static memory mapped device including RAM, ROM and Flash memory.
- Five memory banks with independent configuration
- Eight- and 16-bit wide external memory device supported
- Programmable wait states, up to 16
- Independent byte lane control
- Selective boot memory bank at reset
- Programmable wait states for read and write access

In addition to the address and data buses, the EBI also provides for the following control signals:

- READ. Active low read strobe that is asserted while data is read from the external peripheral.
- WRITE. Active low write strobe that is asserted while data is being written to the external peripheral.
- CS[5:0]. Configurable CS signals.
- BS[1:0]. Active low upper byte-/lower byte-select signals. These are used when the BP is transferring byte-wide data to/from 16-bit peripherals. Signal polarity is programmable.

Note: *The BP always produces a byte address. When 32-bit word data is transferred, A[1:0] bits are always low, that is, set to "0". When 16-bit data half word data is transferred, the A[0] bit is always low, that is, set to "0".*

The EBI read diagram, Figure 3, shows a read from a 16-bit external device. The EBI write diagram, Figure 4, shows a write to a 16-bit external device.

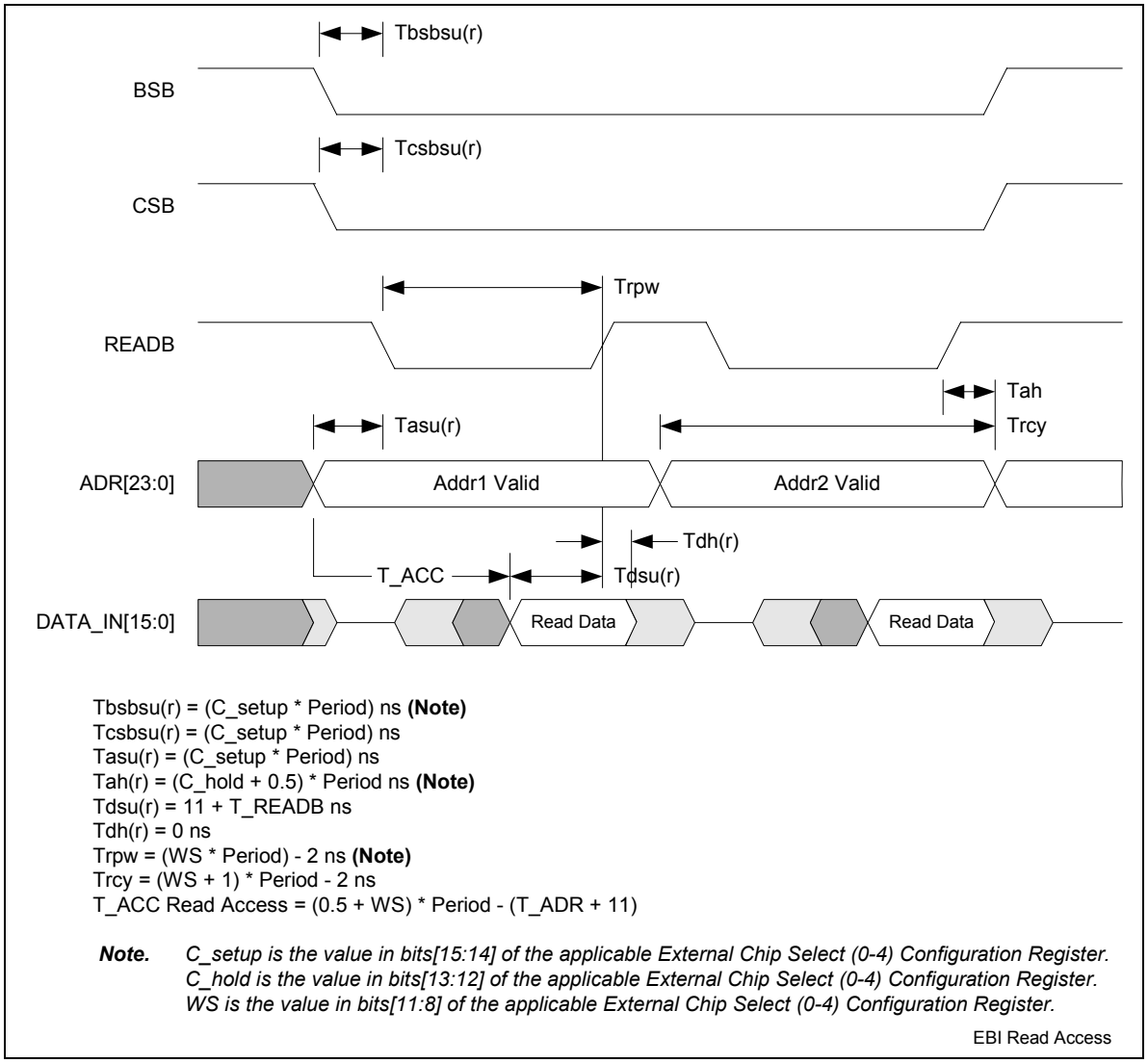
Table 3 provides all EBI timing parameter values.

Chip Select Signals

The BP has five CS signals. Two of these, CS0 and CS1, are on dedicated pins. The other three are multiplexed with GPIO signals. Each CS signal has a configuration register. Register bit functions are shown in Table 5.

ARM register address and default values are specified in Table 6. DSP register address and default values are specified in Table 7.

An example of Worst Case Read Access (T_ACC) Times for two Wait States is shown in Table 4.

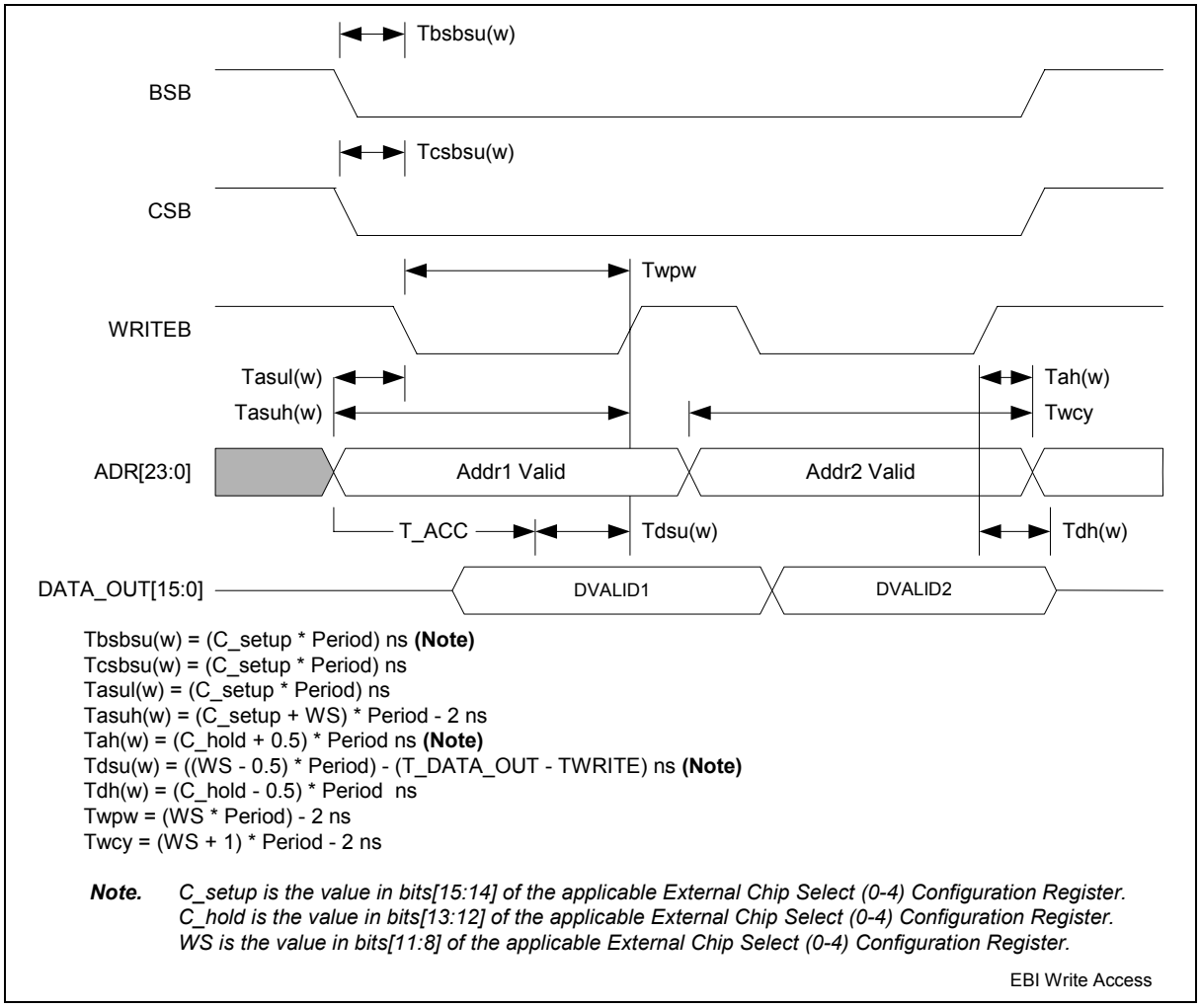


Tbsbsu(r) = (C_setup * Period) ns **(Note)**
Tcsbsu(r) = (C_setup * Period) ns
Tasu(r) = (C_setup * Period) ns
Tah(r) = (C_hold + 0.5) * Period ns **(Note)**
Tdsu(r) = 11 + T_READB ns
Tdh(r) = 0 ns
Trpw = (WS * Period) - 2 ns **(Note)**
Trcy = (WS + 1) * Period - 2 ns
T_ACC Read Access = (0.5 + WS) * Period - (T_ADR + 11)

Note. *C_setup is the value in bits[15:14] of the applicable External Chip Select (0-4) Configuration Register. C_hold is the value in bits[13:12] of the applicable External Chip Select (0-4) Configuration Register. WS is the value in bits[11:8] of the applicable External Chip Select (0-4) Configuration Register.*

EBI Read Access

Figure 3. EBI Read Timing Diagram



Tbsbsu(w) = (C_setup * Period) ns **(Note)**
Tcsbsu(w) = (C_setup * Period) ns
Tasul(w) = (C_setup * Period) ns
Tasuh(w) = (C_setup + WS) * Period - 2 ns
Tah(w) = (C_hold + 0.5) * Period ns **(Note)**
Tdsu(w) = ((WS - 0.5) * Period) - (T_DATA_OUT - TWRITE) ns **(Note)**
Tdh(w) = (C_hold - 0.5) * Period ns
Twpw = (WS * Period) - 2 ns
Twcy = (WS + 1) * Period - 2 ns

Note. *C_setup is the value in bits[15:14] of the applicable External Chip Select (0-4) Configuration Register. C_hold is the value in bits[13:12] of the applicable External Chip Select (0-4) Configuration Register. WS is the value in bits[11:8] of the applicable External Chip Select (0-4) Configuration Register.*

EBI Write Access

Figure 4. EBI Write Timing Diagram

Table 3. EBI Timing Specifications

Parameter	Max Delay
T_ADR	7.8
T_READB	8.8
T_CSB	8.8
T_BSB	8.8
T_DATA_OUT	11 ns
T_WRITEB	9.3 ns
Note. *All delays parameters are for a 10 pF load.*	

Table 4. Example of Worst Case Read Access (T_ACC) Times

	CX805-20	
Frequency	**10 pF Load**	**20 pF Load**
50.7 MHz	77.57	76.27
46.8 MHz	85.69	84.39
39.0 Mhz	106.8	105.5

Table 5. Chip Select Configuration Register

Chip Select Configuration Register Bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Setup		Hold		Wait States				P	Reserved (read as 0)			Size		Zero WS	Enable

Bits 15:14 Control Signal Setup Time
00 == 0.5 Clocks
01 == 1.5 Clocks
10 == 2.5 Clocks
11 == 3.5 Clocks

Bits 13:12 Control Signal Hold Time
00 == 0.5 Clocks
01 == 1.5 Clocks
10 == 2.5 Clocks
11 == 3.5 Clocks

Bits 11:8 Wait States
0000 == 1 ARM Wait State, 1 Cycle Strobe
0001 == 2 ARM Wait State, 2 Cycle Strobe
0010 == 3 ARM Wait State, 3 Cycle Strobe
0011 == 4 ARM Wait State, 4 Cycle Strobe
0100 == 5 ARM Wait State, 5 Cycle Strobe
0101 == 6 ARM Wait State, 6 Cycle Strobe
0110 == 7 ARM Wait State, 7 Cycle Strobe
0111 == 8 ARM Wait State, 8 Cycle Strobe
1000 == 9 ARM Wait State, 9 Cycle Strobe
1001 == 10 ARM Wait State, 10 Cycle Strobe
1010 == 11 ARM Wait State, 11 Cycle Strobe
1011 == 12 ARM Wait State, 12 Cycle Strobe
1100 == 13 ARM Wait State, 13 Cycle Strobe
1101 == 14 ARM Wait State, 14 Cycle Strobe
1110 == 15 ARM Wait State, 15 Cycle Strobe
1111 == 16 ARM Wait State, 16 Cycle Strobe

Bit 7 Chip Select Polarity
1 == Active High
0 == Active Low

Bits 6:4 Reserved – Will Read as 0

Bits 3:2 Width of device using this chips select
00 == 8 bits
01 == 16 bits
10 == 32 bits
11 == Undefined

Bit 1 Reserved, must be set to zero

Bit 0 Chip Select Enable
1 == Enabled
0 == Disabled

Table 6. ARM Register Address Map (1 of 9)

Address (Hex)	Loc Size (bytes) // Init Value	Type	Name	Function
Internal RAM				
0x0000004-0x0002FFF	0x02FFC	R/W	Internal RAM	3K x 32 fast internal RAM implemented (512 Words write protectable)
0x0003000-0x003FFFF	0x3D000	R/W	Available for IRAM expansion	Up to 256 kBytes total internal RAM expansion
Internal Peripherals				
0x0040000-0x005FFFF	0x20000	R/W	Internal Peripherals	All peripherals in the IPB except the DSP Scratchpad
AMBA				
4,0010	2 // x0605	R/W	CS0_CONFIG	EBI CS0 configuration register
4,0012	2 // x0000	R/W	CS1_CONFIG	EBI CS1 configuration register
4,0014	2 // x0000	R/W	CS2_CONFIG	EBI CS2 configuration register
4,0016	2 // x0000	R/W	CS3_CONFIG	EBI CS3 configuration register
4,0018	2 // x0000	R/W	CS4_CONFIG	EBI CS4 configuration register
4,001A	2 // x0004	R/W	AMBA_CTRL	AMBA Control register
4,001C	2 // x0000	R/W	AMBA_STAT	AMBA Status register
4,001E	2 // x0000	R/W	CACHE_CTL	Reserved. Cache Control in CX810
Real Time Clock				
4,0080	2 // xuuuu	R/W	RTC Seconds	Real Time Clock
4,0082	2 // xuuuu	R/W	RTC Minutes	Real Time Clock
4,0084	2 // xuuuu	R/W	RTC Hours	Real Time Clock

Table 6. ARM Register Addresses (2 of 9)

Address (Hex)	Loc Size (bytes) // Init Value	Type	Name	Function
Real Time Clock (continued)				
4,0086	2 // xuuuu	R/W	RTC Days	Real Time Clock
4,0088	2 // xuuuu	R/W	RTC Months	Real Time Clock
4,008A	2 // xuuuu	R/W	RTC Years	Real Time Clock
4,008C	2 // xuuuu	R/W	RTC Control	Real Time Clock
4,008E	2 // xuuuu	R/W	RTC Test	RTC Test Mode Control Register
4,0092	2 // xuuuu	R/W	Alarm Minutes	Alarm
4,0094	2 // xuuuu	R/W	Alarm Hours	Alarm
4,0096	2 // xuuuu	R/W	Alarm Days	Alarm
4,0098	2 // xuuuu	R/W	Alarm Months	Alarm
4,009A	2 // xuuuu	R/W	Alarm Years	Alarm
4,009C	1 // xuuuu	R/W	Interrupt Control	External Interrupt Enable
4,00A0	2 // xuuuu	R/W	Storage Register #0	General Storage Register
4,00A2	2 // xuuuu	R/W	Storage Register	General Storage Register
4,00A4	2 // xuuuu	R/W	Storage Register	General Storage Register
4,00A6	2 // xuuuu	R/W	Storage Register	General Storage Register
4,00A8	2 // xuuuu	R/W	Storage Register	General Storage Register
4,00AA	2 // xuuuu	R/W	Storage Register	General Storage Register
4,00AC	2 // xuuuu	R/W	Storage Register	General Storage Register
4,00AE	2 // xuuuu	R/W	Storage Register	General Storage Register
4,00B0	2 // xuuuu	R/W	Storage Register	General Storage Register
4,00B2	2 // xuuuu	R/W	Storage Register	General Storage Register
4,00B4	2 // xuuuu	R/W	Storage Register	General Storage Register
4,00B6	2 // xuuuu	R/W	Storage Register	General Storage Register
4,00B8	2 // xuuuu	R/W	Storage Register	General Storage Register
4,00BA	2 // xuuuu	R/W	Storage Register	General Storage Register
4,00BC	2 // xuuuu	R/W	Storage Register	General Storage Register
4,00BE	2 // xuuuu	R/W	Storage Register #15	General Storage Register
FIQ and IRQ (Interrupt Controller 1)				
4,0100	4 // x0000, x0000	R/W	INT1_IPR	Interrupt Pending Register
4,0104	4 // x0000, x0000	R/W	INT1_IPOL	Interrupt Polarity Register
4,0108	4 // x0000, x0000	R/W	INT1_IER	Interrupt Enable Register
4,010C	4 // x0000, x0000	R/W	INT1_ISEL	Interrupt Select Register
4,0110	4 //xFFFF, xFFFF	R/W	INT1_ICOND	Interrupt Condition Register
4,0114	4 // x0000, x0000	R/W	INT1_FIQ	FIQ Interrupts Register
4,0118	4 // x0000, x0000	R/W	INT1_IRQ	IRQ Interrupts Register
4,011C	4 // x0000, x0000	R/W	INT1_TEST	Interrupt Test Register
4,0120	2 // x0000	R	INT1_VECT	Interrupt Priority Vector Register
4,0122	4 // x0000, x0000	R	INT1_INTRPTS	Raw Interrupts Register
4,0128	4 // x0000, x0000	R/W	INT1_ITYPE	Interrupt Type Register
CRC				
4,01D0	2 // xFFFF	R/W	CRC_DATA	Writing inputs 8- or 16-bit data to the shift register. Reading reads 16-bit result.
4,01D2	2 // x0000	W	CRC_RST	Writing resets shift register to 0xFFFF

Table 6. ARM Register Addresses (3 of 9)

Address (Hex)	Loc Size (bytes) // Init Value	Type	Name	Function
DEBUG Serial Port				
4,0200	2 // x0000	R/W	Tx Control	Tx Control Registers
4,0202	2 // x0000	R/W	Tx Baud Rate Select	
4,0204	2 // x0000	R/W	Tx Holding	
4,0206	2 // x0000	R/W	Tx Status	
4,0208	2 // x0000	R/W	Rx Control	Rx Control Registers
4,020A	2 // x0000	R/W	Rx Baud Rate Select	
4,020C	2 // x0000	R/W	Rx Holding	
4,020E	2 // x0000	R/W	Rx Status	
4,0210	1 // x0000	R/W	Time Out Delay	Requested Delay
4,0212	2 // x0000	R/W	Delay (Gap) Count	Actual Delay Counted
IrDA Framer				
4,0240	2 // x0000	R/W	Framer Control	
4,0242	2 // x0000	R/W	Status	
4,0244	2 // x0000	R/W	Data Register	
4,0246	2 // x0000	R/W	# of Last Rxd Msg	
4,0248	2 // x0000	R/W	Number of Tx Msgs	
4,024A	2 // x0000	R/W	Xon & Xoff Characters	
4,024C	2 // x0000	R/W	Number of BOFs	
4,024E	2 // x0000	R/W	Interrupt Control/Status	
4,0250	2 // x0000	R/W	Interrupt Control/Status	
4,0252	2 // x0000	R/W	Test Reg for PPM	
4,0254	2 // x0000	R/W	CRC Result LS Half	
4,0256	2 // x0000	R/W	CRC Result MS Half	
4,0258	2 // x0000	R/W	Test/Fault Control	
4,025A	2 // x0000	R/W	Buffer Register	
4,025C	2 // x0000	R/W	PPM Bit Width. PPM	
4,025E	2 // x0000	R/W	Test Shift Reg. PPM	
4,0260	2 // x0000	R/W	Timing Control. PPM	
4,0262	2 // x0000	R/W	Bytes/Msg Rxd	
4,0264	2 // x0000	R/W	Bytes/Msg Txd	
IrDA Framer (cont)				
4,0266	2 // x0000	R/W	Bytes/Last Rxd Msg	
4,0268	2 // x0000	R/W	Chars For BOF/EOF	
GPIOs				
4,0280	2 // x242C	R/W	GRP10_CTL	Groups 1 & 0 Drive and DC Controls
4,0282	2 // x2323	R/W	GRP23_CTL	Groups 3 & 2 Drive and DC Controls
4,0284	2 // xFFFF	R/W	GRPA_CTL	Group A Oen and IE Controls
4,0286	2 // xFFFF	R/W	GRPB_CTL	Group B Oen and IE Controls
4,0288	2 // xFFFF	R/W	GRPC_CTL	Group C Oen and IE Controls
4,028A	2 // xFFFF	R/W	GRPD_CTL	Group D Oen and IE Controls
4,028C	2 // xF0XX	R/W	GRPA_DAT	Group A Data In & Data Out
4,028E	2 // x00XX	R/W	GRPB_DAT	Group B Data In & Data Out
4,0290	2 // x25XX	R/W	GRPC_DAT	Group C Data In & Data Out

Table 6. ARM Register Addresses (4 of 9)

Address (Hex)	Loc Size (bytes) // Init Value	Type	Name	Function
GPIOs (continued)				
4,0292	2 // x00XX	R/W	GRPD_DAT	Group D Data In & Data Out
4,0294	2 // x4800	R/W	CHIP_ID1	Device Identification Registers
4,0296	2 // x0000	R/W	CHIP_ID2	Device Identification Registers
4,2098	2 / xXXXX	R/O	Metal Mod	Metal Mod Revision
I2C/SMBus				
4,0300	2 // x0003	R/W	I2C_Data	I2C/SMBus i/f
4,0302	2 // x0003	R/W	I2C_Clock	I2C/SMBus i/f
Escape Sequence Detect				
4,0320	2 // x0000	R/W	Control	Sets mode of detector
4,0322	2 // x0000	R/W	Status	Detector Status
4,0324	2 // x002B	R/W	ESC Character	Char to use as Escape Sequence Detection
4,0326	2 // x0011	R/W	XON Character	Char to detect as XON
4,0328	2 // x0013	R/W	XOFF Character	Char to detect as XOFF
4,032A	2 // x0000	R/W	ESC Timeout	No. of 20ms periods to count
4,032C	2 // x30B0	R/W	Divider Register	Prescaler divider
SDS IrDA Pulse Shaper				
4,0340	2 // x0000	R/W	IR Control	SDS IR Translator
4,0342	2 // x0000	R/W	IR Baud Generator	
4,0344	2 // x0000	R/W	IR Configuration	
DMA Controller				
4,0420	2 // x0000	R/W	Channel Control	DMA Channel #0
4,0422	2 // x0000	R/W	Channel Status	
4,0424	2 // x0000	R/W	Source LS	
4,0426	2 // x0000	R/W	Source MS	
4,0428	2 // x0000	R/W	Destination LS	
4,042A	2 // x0000	R/W	Destination MS	
4,042C	2 // x0000	R/W	Threshold LS	DMA Channel #0
4,042E	2 // x0000	R/W	Threshold MS	
4,0430	2 // x0000	R/W	End Of Buffer LS	
4,0432	2 // x0000	R/W	End Of Buffer MS	
4,0434	2 // x0000	R/W	Current Source LS	
4,0436	2 // x0000	R/W	Current Source MS	
4,0438	2 // x0000	R/W	Current Destination LS	
4,043A	2 // x0000	R/W	Current Destination MS	
4,0460	2 // x0000	R/W	Channel Control	DMA Channel #1
4,0462	2 // x0000	R/W	Channel Status	
4,0464	2 // x0000	R/W	Source LS	
4,0466	2 // x0000	R/W	Source MS	
4,0468	2 // x0000	R/W	Destination LS	
4,046A	2 // x0000	R/W	Destination MS	
4,046C	2 // x0000	R/W	Threshold LS	
4,046E	2 // x0000	R/W	Threshold MS	
4,0470	2 // x0000	R/W	End Of Buffer LS	

Table 6. ARM Register Addresses (5 of 9)

Address (Hex)	Loc Size (bytes) // Init Value	Type	Name	Function
DMA Controller (continued)				
4,0472	2 // x0000	R/W	End Of Buffer MS	DMA Channel #1
4,0474	2 // x0000	R/W	Current Source LS	
4,0476	2 // x0000	R/W	Current Source MS	
4,0478	2 // x0000	R/W	Current Destination LS	
4,047A	2 // x0000	R/W	Current Destination MS	
4,04A0	2 // x0000	R/W	Channel Control	DMA Channel #2
4,04A2	2 // x0000	R/W	Channel Status	
4,04A4	2 // x0000	R/W	Source LS	
4,04A6	2 // x0000	R/W	Source MS	
4,04A8	2 // x0000	R/W	Destination LS	
4,04AA	2 // x0000	R/W	Destination MS	
4,04AC	2 // x0000	R/W	Threshold LS	
4,04AE	2 // x0000	R/W	Threshold MS	
4,04B0	2 // x0000	R/W	End Of Buffer LS	
4,04B2	2 // x0000	R/W	End Of Buffer MS	
4,04B4	2 // x0000	R/W	Current Source LS	
4,04B6	2 // x0000	R/W	Current Source MS	
4,04B8	2 // x0000	R/W	Current Destination LS	
4,04BA	2 // x0000	R/W	Current Destination MS	
DMA Channels				
4,04E0	2 // x0000	R/W	Channel Control	DMA Channel #3
4,04E2	2 // x0000	R/W	Channel Status	
4,04E4	2 // x0000	R/W	Source LS	
4,04E6	2 // x0000	R/W	Source MS	
4,04E8	2 // x0000	R/W	Destination LS	
4,04EA	2 // x0000	R/W	Destination MS	
4,04EC	2 // x0000	R/W	Threshold LS	
4,04EE	2 // x0000	R/W	Threshold MS	
4,04F0	2 // x0000	R/W	End Of Buffer LS	
4,04F2	2 // x0000	R/W	End Of Buffer MS	
4,04F4	2 // x0000	R/W	Current Source LS	
4,04F6	2 // x0000	R/W	Current Source MS	
4,04F8	2 // x0000	R/W	Current Destination LS	
4,04FA	2 // x0000	R/W	Current Destination MS	
4,0520	2 // x0000	R/W	Channel Control	DMA Channel #4
4,0522	2 // x0000	R/W	Channel Status	
4,0524	2 // x0000	R/W	Source LS	
4,0526	2 // x0000	R/W	Source MS	
4,0528	2 // x0000	R/W	Destination LS	
4,052A	2 // x0000	R/W	Destination MS	
4,052C	2 // x0000	R/W	Threshold LS	
4,052E	2 // x0000	R/W	Threshold MS	
4,0530	2 // x0000	R/W	End Of Buffer LS	

Table 6. ARM Register Addresses (6 of 9)

Address (Hex)	Loc Size (bytes) // Init Value	Type	Name	Function
DMA Channels (continued)				
4,0532	2 // x0000	R/W	End Of Buffer MS	DMA Channel #4
4,0534	2 // x0000	R/W	Current Source LS	
4,0536	2 // x0000	R/W	Current Source MS	
4,0538	2 // x0000	R/W	Current Destination LS	
4,053A	2 // x0000	R/W	Current Destination MS	
4,0560	2 // x0000	R/W	Channel Control	DMA Channel #5
4,0562	2 // x0000	R/W	Channel Status	
4,0564	2 // x0000	R/W	Source LS	
4,0566	2 // x0000	R/W	Source MS	
4,0568	2 // x0000	R/W	Destination LS	
4,056A	2 // x0000	R/W	Destination MS	
4,056C	2 // x0000	R/W	Threshold LS	
4,056E	2 // x0000	R/W	Threshold MS	
4,0570	2 // x0000	R/W	End Of Buffer LS	
4,0572	2 // x0000	R/W	End Of Buffer MS	
4,0574	2 // x0000	R/W	Current Source LS	
4,0576	2 // x0000	R/W	Current Source MS	
4,0578	2 // x0000	R/W	Current Destination LS	
4,057A	2 // x0000	R/W	Current Destination MS	
4,0620	2 // x0000	R/W	DMA Holding Reg LS	DMA Controller Block Test Registers
4,0622	2 // x0000	R/W	DMA Holding Reg MS	
4,0624	2 // x0000	R/W	Peripheral PREQ Test	
4,0626	2 // x0000	R/W	Interrupt Request Test	
4,0628	2 // x0000	R/W	Peripheral Req Status	
Sleep Timer				
4,0700	0x0000	R/W	Control	Sleep & Calibration Block
4,0702	0x0000	R/W	Clock Count	
4,0704	0x0000	R/W	18 Bit Counter LS (16)	
4,0706	0x0000	R/W	18 Bit Counter MS (2)	
4,0708	0x0000	R/W	Sleep Count LS	
4,070A	0x0000	R/W	Sleep Count MS	
Autobaud				
4,0740	0x0000	R/W	Decision Values	16x16bit register set
4,0760	0x0000	R/W	Control	
4,0762	0x0000	R/W	Status	
4,0764	0x0000	R/W	Failure	
4,0766	0x0000	R/W	Characters	
4,0768	0x0000	R/W	Baud Counter	
4,076A	0x0000	R/W	Shift Register	
4,076C	0x0000	R	ASP Control Set	
4,076E	0x0000	R	Block State	

Table 6. ARM Register Addresses (7 of 9)

Address (Hex)	Loc Size (bytes) // Init Value	Type	Name	Function
Keypad I/F				
4,0800	2 // x0000	R/W	Control	GPIO for keypad control
4,0802	2 // x00FF	R/W	Strobe Data In/Out	
4,0804	2 // xFFFF	R/W	Return Data In	
Clock Control Block				
4,0902	2 // x00C0	R/W	Clock Control #1	Clock Enables, Sleep Request
4,0904	2 // x1FC0	R/W	Clock Force #1	Clock Forces
4,0906	2 // x0000	R/W	Clock Control #2	Clock Enables
4,0908	2 // x0000	R/W	Clock Force #2	Clock Forces
4,090A	2 // x0000	R	Clock Request Vector	
4,090C	2 // x00B0	R/W	DSP Configuration	
4,090E	2 // x0000	R/W	DSP Control	
4,0910	2 // x0000	R/W	DPS Interrupt I/F Reg	
4,091E	2 // x0047	R/W	ASIC Mode Control	Major Configuration Control
Pulse Width Modulator				
4,0A00	2 // x0000	R/W	Control	Generator #1
4,0A02	2 // x0000	R/W	Duty Cycle	
4,0A04	2 // x0000	R/W	Divider	
4,0A06	2 // x0000	R/W	Count	
4,0A08	2 // x0000	R/W	Pattern LS	
4,0A0A	2 // x0000	R/W	Pattern MS	
4,0A10	2 // x0000	R/W	Control	Generator #2
4,0A12	2 // x0000	R/W	Duty Cycle	
4,0A14	2 // x0000	R/W	Divider	
4,0A16	2 // x0000	R/W	Count	
4,0A18	2 // x0000	R/W	Pattern LS	
Pulse Width Modulator (cont)				
4,0A1A	2 // x0000	R/W	Pattern MS	
Subscriber Interface Module				
4,0A80	2 // x7420	R/W	Control LS	
4,0A82	2 // x2001	R/W	Control MS	
4,0A84	2 // x1010	R/W	Status	
4,0A86	2 // x0000	R/W	Interrupt Control	
4,0A88	2 // x0000	R/W	Output Buffer	
4,0A8A	2 // x0000	R/W	Input Buffer	
4,0A8C	2 // x0000	R/W	DMA Delay Count	
PLL Control Block				
4,0B00	2 // x433C	R/W	PLL Control	P & N Divide Values, Test Port Enable
4,0B02	2 // x0000	R/W	PLL Dividers	ARM, ARM2x, PLL_ROOT
4,0B04	2 // x0000	R/W	External Control	PLL Enable,

Table 6. ARM Register Addresses (8 of 9)

Address (Hex)	Loc Size (bytes) // Init Value	Type	Name	Function
TEST BLK				
4,0D00	2 //xXXXX	R/W	BOOT Vector	Test Control Bus Vector. S/W AND Reset
4,0D02	2 // x0000	R/W	DSP's GPIO Port	
4,0D04	2 // x0000	R/W	PortA IO Selects	Choose GPIO internal connection
4,0D06	2 // x0000	R/W	PortB IO Selects	
4,0D08	2 // x0044	R/W	PortC IO Selects	
4,0D0A	2 // x0000	R/W	PortD IO Selects	
4,0D0C	2 // x0000	R/W	EmuMux Selection	Before and After test muxes
4,0D0E	2 // x00F5	R/W	TestCtl3 . Misc Ctls	Field Test, Vadjst, etc.
4,0D10	2 // xFF1A	R/W	Shared RAM Control	RAM Assignment and Repair
4,0D12	2 // x0000	R/W	LBIST Pattern Count	LBIST Control Registers
4,0D14	2 // x0000	R/W	LBIST Shift Count	
4,0D16	2 // x0000	R/W	LBIST PRPG Seed LS	
4,0D18	2 // x0000	R/W	LBIST PRPG Seed MS	
SDS Serial Port				
4,1200	2 // x0000	R/W	Tx Control	Tx Control Registers
4,1202	2 // x0000	R/W	Tx Baud Rate Select	
4,1204	2 // x0000	R/W	Tx Holding	
4,1206	2 // x0000	R/W	Tx Status	
4,1208	2 // x0000	R/W	Rx Control	Rx Control Registers
4,120A	2 // x0000	R/W	Rx Baud Rate Select	
4,120C	2 // x0000	R/W	Rx Holding	
4,120E	2 // x0000	R/W	Rx Status	
4,1210	1 // x0000	R/W	Time Out Delay	Requested Delay
4,1212	2 // x0000	R/W	Delay (Gap) Count	Actual Delay Counted
Timers				
4,1300	2 // x0000	R/W	Timer Period LS_16	Timer #1 TPRL
4,1302	2 // x0000	R/W	Timer Period MS_10	Timer #1 TPRM
4,1304	2 // x0000	R/W	Period Holding Reg_16	Timer #1 PHRL
4,1306	2 // x0000	R/W	Period Holding Reg_10	Timer #1 PHRM
4,1308	2 // xFFFF	R/W	Timer Count_16	Timer #1 Count Value TIML (LS 16 bits)
4,130A	2 // xFFFF	R/W	Timer Count_10	Timer #1 Count Value TIMM (MS 10 bits)
4,130C	2 // x0008	R/W	Timer Control	Timer #1 Timer Control Register
4,1400	2 // x0000	R/W	Timer Period LS_16	Timer #2 TPRL
4,1402	2 // x0000	R/W	Timer Period MS_10	Timer #2 TPRM
4,1404	2 // x0000	R/W	Period Holding Reg_16	Timer #2 PHRL
4,1406	2 // x0000	R/W	Period Holding Reg_10	Timer #2 PHRM
4,1408	2 // xFFFF	R/W	Timer Count_16	Timer #2 Count Value TIML (LS 16 bits)
4,140A	2 // xFFFF	R/W	Timer Count_10	Timer #2 Count Value TIMM (MS 10 bits)
4,140C	2 // x0008	R/W	Timer Control	Timer #2 Timer Control Register

Table 6. ARM Register Addresses (9 of 9)

Address (Hex)	Loc Size (bytes) // Init Value	Type	Name	Function
GENDEX				
4,1500	2 // x0000	R/W	Control	
4,1502	2 // x0000	R/W	Status	
4,1504	2 // x0000	R/W	Key Word #1	
4,1506	2 // x0000	R/W	Key Word #2	
4,1508	2 // x0000	R/W	Key Word #3	
4,150A	2 // x0000	R/W	Key Word #4	
4,150C	2 // x0000	R/W	Input Word #1	
4,150E	2 // x0000	R/W	Input Word #2	
4,1510	2 // x0000	R/W	Frame Length	
4,1512	2 // x0000	R/W	NCRC	
4,1514	2 // x0000	R/W	Offset	
4,1516	2 // x0000	R/W	CRC LS	
4,1518	2 // x0000	R/W	CRC MS	
DBG IR Translator				
4,1600	2 // x0000	R/W	IR Control	DBG IR Translator
4,1602	2 // x0000	R/W	IR Baud Generator	
4,1604	2 // x0000	R/W	IR Configuration	
DSP Scratchpad (Internal)				
0x006,0000	0x07FF	R/W	DSP Scratchpad	Internal DSP ScratchPad
DSP Patch RAM				
0x006,1000	0x1FFF	R/W	Only available in Test	Accessable only when DSP is in Reset
GENDEX				
0x006,3000	0x0007	R/W	GPRS Data Port. I/O	32, 16, 8 bit transfers
Cache				
0x006,4000	0x07FF	R/W	Cache Data RAM #1	512x32
0x006,4800	0x07FF	R/W	Cache Data RAM #2	512x32
0x006,5000	0x007F	R/W	Cache Tag RAM #1	128x24
0x006,5080	0x007F	R/W	Cache Tag RAM #2	128x24
0x006,6000	0x000F	R/W	Cache Control	Available on APB Control Port (Preferred)
Internal ROM				
0x008,0000	0x3FFF	R/W	IROM Space	4kx32
DSP DDB (Full Memory Map of Data Space)				
0x010,0000	0x2,0000	R/W	Only available in Test	Access-able only when DSP is in Reset
External Chip Selects				
0x0200000-0x02FFFFF	0x100000	R/W	External SRAM	Up to 1MByte of external expansion (CS1)
0x0300000-0x03FFFFF	0x100000	R/W	Spare	Up to 1MByte of external expansion (CS2)
0x0400000-0x07FFFFF	0x400000	R/W	External FLASH	Up to 4MByte of external expansion (CS0)
0x0800000-0x0BFFFFF	0x400000	R/W	Spare	Up to 4MByte of external expansion (CS4)
0x0C00000-0x0FFFFFF	0x400000	R/W	Spare	Up to 4MByte of external expansion (CS3)

Table 7. DSP Register Addresses (1 of 4)

Address (Hex)	Loc Size (bytes) // Init Value	Type	Name	Function
DSP Control Block				
C000	2 // x0000	R/W	DSP IRQ Control	
C002	2 // x0000	R/W	BOOT Vector	
C004	2 // x0000	R/W	CoProcessor Select	
C006	2 // x0000	R/W	Clock Control	
C008	2 // x0000	R/W	DMA0 Threshold	Receive Port Channel DMA #0
C00A	2 // x0000	R/W	DMA0 Base	
C00C	2 // x0000	R/W	DMA1 Threshold	Receive Port Channel DMA #1
C00E	2 // x0000	R/W	DMA1 Base	
C010	2 // x0000	R/W	DMA Word Data	Last Data (Test)
C012	2 // x0000	R/W	DMA Address	Last Address (Test)
Interrupt Controller				
C100	2 // x0000	R/W	Interrupt Pending (IPR)	Reading gives interrupts that are pending. Writing clears bits; 1 = Clear, 0 = no change
C102	2 // x0000	R/W		
C104	2 // x0000	R/W	Interrupt Polarity (IPOL)	Set the polarity of all interrupts 1 = Asserted True
C106	2 // x0000	R/W		
C108	2 // x0000, x0000	R/W	Interrupt Enable (IER)	Enables the corresponding interrupt source 1 = Enabled
C10A	2 // x0000	R/W		
C10C	2 // x0000, x0000	R/W	Interrupt Select (INTSEL)	Directs a given interrupt to the IRQ (= 0) or the FIQ (= 1) input to the ARM.
C10E	2 // x0000	R/W		
C110	2 //xFFFF	R/W	Interrupt Condition (ICOND)	Detect IRQ only if Bus Clock is Running (= 1), else Asynchronously (= 0).
C112	2 // xFFFF	R/W		
C114	2 // x0000	R/W	FIQ Interrupts	32 bits of IRQs that are enabled to ARM's nFIQ. Status Only.
C116	2 // x0000	R/W		
C118	2 // x0000	R/W	IRQ Interrupts	32 bits of IRQs that are enabled to ARM's nIRQ. Status Only.
C11A	2 // x0000	R/W		
C11C	2 // x0000	R/W	Interrupt Test Control	Set IPR bit for test. 1 = Set, Bit wise
C11E	2 // x0000	R/W		
C120	2 // x0000	R	IRQ Jump Vector	5-bit field of Jump Vector
C128	2 // x0000	R/W	Interrupt Type Select (ITYPE)	Edge sensitive (= 1) or Level sensitive (= 0)
C12A	2 // x0000	R/W		

Table 7. DSP Register Addresses (2 of 4)

Address (Hex)	Loc Size (bytes) // Init Value	Type	Name	Function
Serial Port #1 (Receive)				
C300	2 // x0000	R/W	Tx Control	Tx Control Registers
C302	2 // x0000	R/W	Tx Baud Rate Select	
C304	2 // x0000	R/W	Tx Holding	
C306	2 // x0000	R/W	Tx Status	
C308	2 // x0000	R/W	Rx Control	Rx Control Registers
C30A	2 // x0000	R/W	Rx Baud Rate Select	
C30C	2 // x0000	R/W	Rx Holding	
C30E	2 // x0000	R/W	Rx Status	
C310	1 // x0000	R/W	Time Out Delay	Requested Delay
C312	2 // x0000	R/W	Delay (Gap) Count	Actual Delay Counted
Serial Port #2 (Control)				
C400	2 // x0000	R/W	Tx Control	Tx Control Registers
C402	2 // x0000	R/W	Tx Baud Rate Select	
C404	2 // x0000	R/W	Tx Holding	
C406	2 // x0000	R/W	Tx Status	
C408	2 // x0000	R/W	Rx Control	Rx Control Registers
C40A	2 // x0000	R/W	Rx Baud Rate Select	
C40C	2 // x0000	R/W	Rx Holding	
C40E	2 // x0000	R/W	Rx Status	
C410	1 // x0000	R/W	Time Out Delay	Requested Delay
C412	2 // x0000	R/W	Delay (Gap) Count	Actual Delay Counted
Serial Port #3 (Codec)				
C500	2 // x0000	R/W	Tx Control	Tx Control Registers
C502	2 // x0000	R/W	Tx Baud Rate Select	
C504	2 // x0000	R/W	Tx Holding	
C506	2 // x0000	R/W	Tx Status	
C508	2 // x0000	R/W	Rx Control	Rx Control Registers
C50A	2 // x0000	R/W	Rx Baud Rate Select	
C50C	2 // x0000	R/W	Rx Holding	
C50E	2 // x0000	R/W	Rx Status	
C510	1 // x0000	R/W	Time Out Delay	Requested Delay
C512	2 // x0000	R/W	Delay (Gap) Count	Actual Delay Counted
Serial Port #4 (DAI)				
C600	2 // x0000	R/W	Tx Control	Tx Control Registers
C602	2 // x0000	R/W	Tx Baud Rate Select	
C604	2 // x0000	R/W	Tx Holding	
C606	2 // x0000	R/W	Tx Status	
C608	2 // x0000	R/W	Rx Control	Rx Control Registers
C60A	2 // x0000	R/W	Rx Baud Rate Select	
C60C	2 // x0000	R/W	Rx Holding	
C60E	2 // x0000	R/W	Rx Status	

Table 7. DSP Register Addresses (3 of 4)

Address (Hex)	Loc Size (bytes) // Init Value	Type	Name	Function
C610	1 // x0000	R/W	Time Out Delay	Requested Delay
C612	2 // x0000	R/W	Delay (Gap) Count	Actual Delay Counted
Timer				
C700	2 // x0000	R/W	Timer Period LS_16	TPRL
C702	2 // x0000	R/W	Timer Period MS_10	TPRM
C704	2 // x0000	R/W	Period Holding Reg_16	PHRL
C706	2 // x0000	R/W	Period Holding Reg_10	PHRM
C708	2 // xFFFF	R/W	Timer Count_16	Count Value TIML (LS 16 bits)
C70A	2 // xFFFF	R/W	Timer Count_10	Count Value TIMM (MS 10 bits)
C70C	2 // x0008	R/W	Timer Control	Timer Control Register
Viterbix				
C900	2 // x0000	R/W	Control	
C902	2 // x0000	R/W	Status	
C904	2 // x0000	R/W	Configuration	
C906	2 // x0000	R/W	RAM Address	
C908	2 // x0000	R/W	States	
C90A	2 // x0000	R/W	Number of Symbols	
C90C	2 // x0000	R/W	Polynomial #1	
C90E	2 // x0000	R/W	Polynomial #2	
C910	2 // x0000	R/W	Polynomial #3	
C912	2 // x0000	R/W	Polynomial #4	
C914	2 // x0000	R/W	Polynomial #5	
C916	2 // x0000	R/W	Polynomial #6	
Trap and Patch				
CA00	2 // x0000	R/W	Patch Address #0	Address at which to Trap
CA02	2 // x0000	R/W	Patch Address #1	
CA04	2 // x0000	R/W	Patch Address #2	
CA06	2 // x0000	R/W	Patch Address #3	
CA08	2 // x0000	R/W	Patch Address #4	
CA0A	2 // x0000	R/W	Patch Address #5	
CA0C	2 // x0000	R/W	Patch Address #6	
CA0E	2 // x0000	R/W	Patch Address #7	
CA10	2 // x0000	R/W	Patch Address #8	
CA12	2 // x0000	R/W	Patch Address #9	
CA14	2 // x0000	R/W	Patch Address #A	
CA16	2 // x0000	R/W	Patch Address #B	
CA18	2 // x0000	R/W	Patch Address #C	
CA1A	2 // x0000	R/W	Patch Address #D	
CA1C	2 // x0000	R/W	Patch Address #E	
CA1E	2 // x0000	R/W	Patch Address #F	
CA28	2 // x0000	R/W	DataAdr Trap	Alternate Traps
CA2A	2 // x0000	R/W	Opcode LS Trap	<<Not Available>>
CA2C	2 // x0000	R/W	Opcode MS Trap	<<Not Available>>

Table 7. DSP Register Addresses (4 of 4)

Address (Hex)	Loc Size (bytes) // Init Value	Type	Name	Function
Trap and Patch (cont)				
CA2E	2 // x0000	R/W	Control Traps	
CA30	2 // x0000	R	Status	
CA32	2 // x0000	R/W	Patch Enables A	Patch Enables
CA34	2 // x0000	R/W	Patch Enables B	
CA36	2 // x0000	R	Last PC Addr LS	Debug Only
CA38	2 // x0000	R	Last PC Addr MS	
Cidex				
CB00	2 // x0000	R/W	Control	Block Registers
CB02	2 // x0000	R/W	Status	
CB04	2 // x0000	R/W	Key #1	
CB06	2 // x0000	R/W	Key #2	
CB08	2 // x0000	R/W	Key #3	
CB0A	2 // x0000	R/W	Key #4	
CB0C	2 // x0000	R/W	TDMA #1	
CB0E	2 // x0000	R/W	TDMA #2	
CB10	2 // x0000	R/W	Cipher Reg Adr #1	First Block
CB12	2 // x0000	R/W	Cipher Reg Adr #2	
CB14	2 // x0000	R/W	Cipher Reg Adr #3	
CB16	2 // x0000	R/W	Cipher Reg Adr #4	
CB18	2 // x0000	R/W	Cipher Reg Adr #5	
CB1A	2 // x0000	R/W	Cipher Reg Adr #6	
CB1C	2 // x0000	R/W	Cipher Reg Adr #7	
CB1E	2 // x0000	R/W	Cipher Reg Adr #8	
CB20	2 // x0000	R/W	Cipher Reg Adr #1	Second Block
CB22	2 // x0000	R/W	Cipher Reg Adr #2	
CB24	2 // x0000	R/W	Cipher Reg Adr #3	
CB26	2 // x0000	R/W	Cipher Reg Adr #4	
CB28	2 // x0000	R/W	Cipher Reg Adr #5	
CB2A	2 // x0000	R/W	Cipher Reg Adr #6	
CB2C	2 // x0000	R/W	Cipher Reg Adr #7	
CB2E	2 // x0000	R/W	Cipher Reg Adr #8	

Byte Select Signals

The byte select (BS) signals, BS[1:0], are used to transfer byte-wide data to and from 16-bit peripherals. Both of these signals are active low.

- When BS[0] is asserted, it indicates that data is on bits D[7:0].
- If BS[1] is asserted, data is on bits D[15:8].

During write operations to 16-bit peripherals, the BS signals must be connected to the corresponding pins on the peripheral. These signals allow each 8-bit half of the 16-bit peripheral register to be written to independently. This is required since the ARM compiler may generate two byte transactions when accessing a 16-bit peripheral instead of a single half word (16-bit) transfer.

Byte select signal polarity is programmable using SIU Configuration Register bit[15].

- If bit[15] is set to "0", the signals are active low.
- If bit[15] is set to "1", the bits are active high.

Clock Generation and Phase Locked Loop

The BP clock generation circuitry takes a 3.9 MHz square wave system clock input, buffers it, and routes it to the internal peripherals. Each peripheral has a dedicated clock-enable signal so that the clock signal can be turned off when the peripheral is not in use.

The 3.9 MHz signal is also routed to the Phase Lock Loop (PLL) circuitry, which generates both the ARM and DSP clock signals.

Clock Enables

Each of the device circuitry blocks has a dedicated clock enable signal. This allows the clock signal to the circuitry block to be turned off when not in use. The Clock Control Register # 1 contents control clock enable signals. If a particular bit is set to "1," the clock to the associated block is turned on; if the bit is set to "0," the clock is turned off. Table 8 describes the function of each bit in this register.

Table 8. Clock Control Register Functions

Bit	Block Controlled	Bit	Block Controlled
[15]	IrDA/Escape Sequence	[7]	PTGA
[14]	DSP	[6]	CRC
[13]	DMA Controller	[5]	Timer B
[12]	DMA Controlled	[4]	PWM
[11]	Autobaud	[3]	SIM
[10]	SDS Port	[2]	SLP
[9]	Debug Port	[1]	SIU
[8]	PTGB	[0]	ARM Core

If a "0" is written to a specific bit, the associated clock goes low at the next system clock high-to-low transition. The clock stays low until it is enabled again.

If a "1" is written to a specific bit, the associated clock is turned on at the next system clock low-to-high transition.

Clock Control Register address and default settings are specified in Table 6.

PLL Operation

A PLL functional block diagram is shown in Figure 5.

The system clock input, 3.9 MHz, is divided down by the division factor, P. This factor is a 2-bit number with a value of 2. The output from this divider is input to the PLL block, which generates an output at N times the input frequency, where N is the multiplying factor. The divider or multiplier values are one greater than the programmed value in P or N bits. The PLL output is input to the DSP core. The PLL output is also divided down by a factor, M, to generate the ARM clock. The value of M is 2. In normal operation, DSP Core runs at 48 MHz, while the ARM Core runs at 24 MHz.

Detailed PLL Control Register setup configuration data are in Table 9.

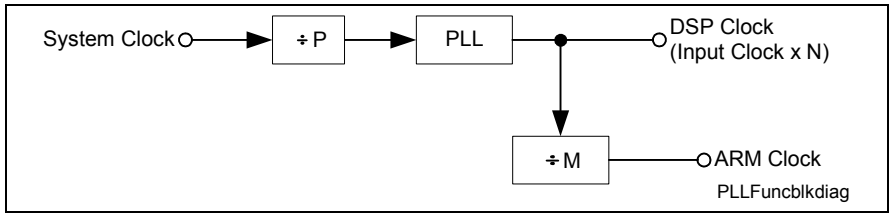


Figure 5. PLL Functional Block Diagram

Table 9. PLL Control Register Data

PLL Control Register Bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEN	N/A	N/A	N/A	N/A	N/A	P-Divider		N/A	Q-Divider		N-Divider				

Bit[15] 1 = PLL Test Output Enable
0 = PLL Test Output Disabled
Bit[14:10] Built, But Not Used.
Bit[9:8] P Divider Value. This value is used to Divide the Input Frequency to PLL

9	8	P Divide
1	1	4
1	0	3
0	1	2
0	0	1

Bit[7] Built, but not Used

Bit[6:5] Q Divider Value. This value is used, along with the N value, to divide the PLL output frequency before it is fed back into the PLL as the Feedback Reference. It is also used to select the PLL Bypass Mode.

6	5	Q Divide
1	1	PLL Bypass
1	0	4
0	1	2
0	0	1

Bit 4. 0 N Divider Value. This value is used, along with the Q value, to divide the PLL output frequency before it is fed back into the PLL as the Feedback Reference. The N divide value is one greater than the 5-bit value set in this field, for example, a zero value divides by 1, a 31 value divides by 32.

ARM Interrupt Controller

The ARM core can handle two interrupts:

- Fast Interrupt Request (FIQ)
- Interrupt Request (IRQ)

The FIQ has a higher priority than the IRQ. The IRQ is masked when an FIQ sequence is entered. In the case of an FIQ interrupt, fewer registers are required to be saved to memory. Therefore, switching into the interrupt handler is slightly faster.

All possible interrupt sources, both internal and external, are routed to the Interrupt Controller, which generates either the FIQ or IRQ interrupt. There are two independent Interrupt Controller block inside the CX805. The first Interrupt Controller has the following features:

- Supports 32 fixed interrupt sources
- Programmable level or edge sensitivity for each interrupt source
- Programmable input polarity for each interrupt source
- Provides a mapping to the FIQ or IRQ outputs
- Programmable interrupt enables mask
- Programmable test interrupts enabling embedded verification and test
- Six-bit Priority Encoded Interrupt Vector generated as output.

Interrupt Controller Registers

The address and default settings for the Interrupt Controller Registers are specified in Table 6.

Interrupt Pending Register. All interrupt sources are latched into the Interrupt Pending Register. When an interrupt is latched into this register, the bit remains set to "1" until the interrupt source has disappeared and the ARM clears the bit. The source of each of the bits in the register is specified in Table 10.

Interrupt Select Register. Every enabled interrupt source can generate either an FIQ or IRQ interrupt to the ARM core. The Interrupt Select Register contains a bit for each possible interrupt source.

- If the associated bit is set to "1", an FIQ interrupt is generated when an interrupt occurs and the interrupt is enabled.
- Conversely, if the bit is set to "0," an IRQ interrupt is generated when an interrupt occurs and the interrupt is enabled.

The Interrupt Select Register bits have the same mapping to the interrupt sources as the Interrupt Pending Register, Table 10.

Interrupt Enable Register. The Interrupt Enable Register contains a corresponding bit for each possible interrupt source.

- If the bit is set to "1," and an interrupt occurs, an interrupt is sent to the ARM. Either an FIQ or IRQ interrupt is generated, depending on the status of the associated interrupt bit in the Interrupt Select Register.
- If the bit is set to "0," the interrupt is disabled.

The Interrupt Enable Register bits have the same mapping to the interrupt sources as the Interrupt Pending Register, Table 10.

Interrupt Type Register. This register defines the edge or level triggered type interrupts.

- If the bit is set to "1", the corresponding interrupt is edge sensitive.
- Otherwise a level is detected. An edge sensitive interrupt can be at a "1" or "0" level without causing an interrupt request. Only the actual edge causes an interrupt request.

Table 10. Interrupt Pending Register Sources

Cntlr	Intrpt #	Interrupt Name	Cntlr	Intrpt #	Interrupt Name
			Intc1	18	Debug USART PIO Data Request
			Intc1	17	GPIO port B bit 0
			Intc1	16	SDS USART PIO Data Ready
			Intc1	15	SDS USART Wakeup Interrupt
			Intc1	14	SDS USART PIO Data Request
			Intc1	13	Gendex Interrupt
Intc1	31	DSP to ARM SUM Interrupt	Intc1	12	Ground
Intc1	30	Frame Tick Interrupt	Intc1	11	DMA Channel 5 Interrupt
Intc1	29	IrDA Framer Interrupt	Intc1	10	DMA Channel 4 Interrupt
Intc1	28	GPIO port B bit 6	Intc1	9	DMA Channel 3 Interrupt
Intc1	27	GPIO port B bit 1	Intc1	8	DMA Channel 2 Interrupt
Intc1	26	DSP to ARM Interrupt	Intc1	7	DMA Channel 1 Interrupt
Intc1	25	Sleep/Cal Wakeup Interrupt	Intc1	6	DMA Channel 0 Interrupt
Intc1	24	Keydetect Interrupt	Intc1	5	GPIO port B bit 2
Intc1	23	GPIO port B bit 4)	Intc1	4	Sleep/Cal Calibrate Interrupt
Intc1	22	GPIO port B bit 5	Intc1	3	SIM Interrupt
Intc1	21	GPIO port B bit 3	Intc1	2	RTC Alarm Interrupt
Intc1	20	Debug USART PIO Data Ready	Intc1	1	Timer 2 Interrupt
Intc1	19	Autobaud Interrupt	Intc1	0	Timer 1 Interrupt

Interrupt Polarity Register. The polarity of all external interrupts is selected by writing to the appropriate bit in the Interrupt Polarity Register.

- If the bit is set to "1", an interrupt is generated either on the rising edge for edge mode or on the high level for level mode.
- If the bit is set to "0", an interrupt is generated either on the falling edge for edge mode or on the low level for level mode.

Caution: *Care must be taken since the act of altering the bit could result in the generation of an interrupt edge. This potential hazard can be avoided by using software to disable the interrupt source when the polarity bit is changed.*

The External Interrupt Polarity Register bits have the same mapping to the interrupt sources as the Interrupt Pending Register, Table 10. For internally generated interrupts, the associated bits in this register are unused.

Interrupt Condition Register. This register defines the corresponding interrupts to be synchronous or asynchronous detected.

- If the bit is set to "1", the interrupt is synchronous detected.
- Otherwise, an asynchronous is assumed.

FIQ Interrupt Register. The FIQ Interrupt Register contains bits for all the possible interrupt sources. The FIQ Interrupt Register bits have the same mapping to the interrupt sources as the Interrupt Pending Register, Table 10. If a bit for a particular interrupt is set to 1, the following conditions apply to that interrupt:

- The interrupt has occurred.
- The interrupt is enabled.

- The interrupt is set to generate an FIQ interrupt.

If the bit is set to 0, at least one of the conditions listed above is not met.

IRQ Interrupt Register. The IRQ Interrupt Register contains bits for all the possible interrupt sources. The IRQ Interrupt Register bits have the same mapping to the interrupt sources as the Interrupt Pending Register, Table 10.

- If a bit for a particular interrupt is set to "1", the following conditions apply to that interrupt:
  - The interrupt has occurred
  - The interrupt is enabled
  - The interrupt is set to generate an IRQ interrupt
- If the bit is set to "0", at least one of the conditions listed above is not met.

Timers

There are two 26-bit general purpose counters/timers used to generate time related interrupts to the ARM.

- Timer A. Used to generate timeouts related to the SIM interface. This is required by the ETSI GSM specifications. Timer A uses either the SIM system clock or the SIM Elementary Time Unit (ETU) clock as an input clock. The output from Timer A is input to the IC.
- Timer B. Used for general-purpose timing. Its input clock can be the system clock, 3.9 MHz, or the ARM clock. Timer A output is input to the Interrupt Controller as timb_irq.

Timer A and Timer B each consist of a Latch Register and a Counter Register. The Counter Register is loaded from the Latch Register, and the timer counts down the contents of the Counter Register. When the Counter Register contents reach 0, the interrupt is generated.

Each of the timers has a dedicated Timer Control Register. The contents of this register determine the configuration of the timer. The function of each bit in the register is provided in Table 11.

Table 11. Timer Mode Register Functions

Timer Control Register Data																
offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0C	Reserved									ESN	TLS	INE	TMS	CKS	MOD	

Read value after reset = 0x0008

Bit[5] TLS (Timer Load Select)
- 0 = Data written to TPRM/TPRL or TIMM/TIML is not used by the timer until both MS and LS halves are loaded. This ensures the timer gets loaded with the entire 26-bit at the same time so it is not necessary to stop the timer when changing TPR or TIM values.
- 1 = Data written to TPRM/TPRL or TIMM/TIML will be loaded into the corresponding bits in the timer immediately after synchronization. This mode can be used if only half of the TPR or TIM needs to be changed.

Bit[4] INE (Interrupt Enable)
- 0 = disable interrupt output generation
- 1 = enable interrupt output generation

Bit[3] TMS (Timer Stop)
- 0 = normal condition, timer counts
- 1 = stops timer clock.

Bit[2] CKS (Clock Select) *
- 0 = timer uses input clock CLKIN1
- 1 = timer uses input clock CLKIN2

Bit[1:0] MOD (Mode) *

2 bits select the timer operating mode as follows:
- 00 = Interval Timer
- 01 = Pulse Generator
- 10 = Event Counter
- 11 = Pulse Width Measurement

UART Ports

The BP device provides two asynchronous, full duplex serial ports:

- Debug Port. This port is used to output protocol stack signaling information and for remote control of the handset.
- Serial Data Services (SDS) Port. This port is used to communicate with a data terminal or PC.

Both ports can operate at baud rates up to 460.0 kbps. Port timing is generated by two precision timers which, in turn, derive their timing from the 3.9 MHz system clock.

The timing of both ports is RS-232-compatible, and only requires external voltage translation circuitry to interface either of the ports to an RS-232 interface on a PC. Each UART port can support the following functions:

- Independent transmitter and receiver for full duplex operation
- Independent baud rate generators
- Transfer 7- or 8-data-bit, 1 start-bit, 1 or 2 stop-bits, even/odd/mark, or no-parity in asynchronous mode
- Transfer 13- or 16-bit data with one or zero start-bit but no stop-bit or parity-bit in synchronous mode
- PIO or DMA mode data transfer
- Maskable interrupts

Hardware flow control using Clear-to-Send (CTS) and Request-to-Send (RTS) can be emulated by using two pins on Port E with interrupt capability.

There are eighteen registers on each UART port. The address and default values for each UART port registers are specified in Table 6.

Program the Transmit Configuration, Table 12, and the Receive Configuration Register, Table 13 to configure UART port operation.

The UART transmitter section operational parameters are setup in this register. It is possible to change the word format on a byte-by-byte basis by appropriate writes to the configuration register. First, the configuration register is written, then the data byte is loaded. This would only work in PIO mode.

The UART Receiver Configuration Register is used to set up the receiver part of the UART.

Note. *Care must be taken in full duplex modes when changing the message format.*

Table 12. UART Transmit Configuration Register

UART Transmit Configuration Register																
Data Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	Startf	Mstr	FE	EOT	MS[1:0]		IM	IP	HE	Stop	PT[1:0]		Par	BC	PIO	TE
Write	Startf	Mstr	FE	EOT	MS[1:0]		IM	IP	HE	Stop	PT[1:0]		Par	BC	PIO	TE
BnRES	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All defined bits in this register are Read/Write

Bit[15] = Startbit False (Startf)
- 0 = Start bit transmitted during synchronous transfer (default)
- 1 = No Start bit transmitted during synchronous transfer

Note. *This bit is only valid for synchronous transfers and has no affect during asynchronous transfers.*

Bit[14] = Master Synch Transfer (MSTR)
- 0 = Do not master synchronous transfer (default).
- 1 = Allow ASP to master synchronous transfer, and generate the required synchronous clock

Bit[13] = Force Echo (FE)

This bit is used only in ASP Autobaud and is not functional in a non-Autobaud ASP. This bit allows the user to relinquish control to the Autobaud block when asserted with the TE bit.

Bit[12] = Xmt end (EOT)
- 0 = data transferring is ongoing
- 1 = Indicates end of data transmission to external device, it has to be asserted before the last word is written to XMT holding register. This signal is level based, not a pulse.

Bit[11:10] = Data Format (MS[1:0])
- 00 = Asynchronous 8-bit data mode
- 01 = Synchronous 16-bit data mode
- 10 = Synchronous 13-bit data mode
- 11 = Asynchronous 16-bit data mode

Bit[9] = Interrupt Mode (IM)
- 0 = generate an interrupt when the transmission holding register (THR) is empty
- 1 = generate interrupt when both the THR and the Transmit Shift Registers (TSR) are empty

Bit[8] = Interface polarity (IP)
- 0 = no inversion (default)
- 1 = The ASP_DOUT and ASP_DTRF are inverted.

Bit[7] = Halt Enable (HE)
- 0 = Halt mode is disabled (default)
- 1 = Halt mode enabled

Bit[6] = Stop Bit Count (STOP)
- 0 = 1 stop bit (default)
- 1 = 2 stop bits

Bit[5:4] = Parity Type (PT[1:0])
- 00 = even parity(default)
- 01 = odd parity
- 10 = space parity (0)
- 11 = mark parity (1)

Bit[3] = Parity Enable (Par)
- 0 = no parity. (default)
- 1 = parity enabled

Bit[2] = Bit Count (BC)
- 0 = 8 bits per character (default)
- 1 = 7 bits per character

Bit[1] = PIO mode (PIO)
- 0 = DMA mode is selected (default)
- 1 = PIO mode is selected

Bit[0] = Transmit Enable (TE)
- 0 = disable transmit circuitry (default)
- 1 = enable transmit

Table 13. UART Receiver Configuration Register

Data Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	Startf	Echo	WE	TIE	TOE	XW	IW	IP	-	Stop	PT[1:0]		Par	BC	PIO	RE
Write	Startf	Echo	WE	TIE	TOE	XW	IW	IP		Stop	PT[1:0]		Par	BC	PIO	RE
BNRES	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Read/Write
Read. Bit[15] = Startbit False (Startf)
0 = Start bit transmitted during synchronous transfer (default).
1 = No Start bit transmitted during synchronous transfer.
Note. *This bit is only valid for synchronous transfers and has no affect during asynchronous transfers.*
Bit[14] = Echo Mode (ECHO)
0 = Normal Operation (default)
1 = Wrap ASP DIN to ASP DOUT
Bit[13] = Wake up Enable (WE)
0 = Wake up interrupt is disabled (default)
1 = Wake up interrupt is enabled
Bit[12] = Time Out Int Enable (TIE)
0 = Time out interrupt is disabled (default)
1 = Time out interrupt is enabled
Bit[11] = Time Out Enable (TOE)
0 = Time out logic is disabled (default)
1 = Time out logic is enabled
Bit[10] = External Wrap (XW)
0 = normal mode (default)
1 = test enabled
Bit[9] = Internal Wrap (IW)
0 = normal mode (default)
1 = test enabled.
Bit[8] = Interface Polarity (IP)
0 = no inversion (default)
1 = invert the ASP_DIN and ASP_TDRF signals.
Bit[7] = NA
Bit[6] = Stop Bit Count (STOP)
0 = 1 stop bit (default)
1 = 2 stop bits
Bit[5:4] = Parity Type (PT[1:0])
00 = even parity (default)
01 = odd parity
10 = space parity (0)
11 = mark parity (1)
Bit[3] = Parity Enable (PAR)
0 = no parity. (default)
1 = parity enabled
Bit[2] = Bit Count (BC)
0 = 8 bits per character (default)
1 = 7 bits per character.
Bit[1] = PIO mode (PIO)
0 = DMA mode is selected (default)
1 = PIO mode is selected.
Bit[0] = Receive Enable. (RE)
0 = disable receiver circuitry
1 = enable receiver logic
BNRES Bit[15:0] = 0

The baud rate for each UART port is selected by programming the Baud Rate Register with the divisor value. The accuracy of the baud rate is shown in Table 14.

Debug Port

There are two Debug Port signals:

- DEBUG_TX (transmit data output from the device)
- DEBUG_RX (receive data input to the device)

The serial port sends signaling information from the protocol stack to monitor handset operation on a PC. The serial port receives data from a PC to allow remote control of the handset by the PC, for example, originate a call, terminate a call.

Serial Data Services Port

There are two SDS Port signals:

- SDS_TX (transmit data)
- SDS_RX (receive data)

PTG A generates SDS timing. The clock to the interface circuitry is normally off to save power. When a start bit is seen on the SDS_RX signal or a byte is written to the port output buffer, the clock is turned on.

Infrared Data Adapter Pulse Shaper

The IrDA Pulse Shaper allows an IrDA-compatible transceiver to be connected directly to the SDS pins on the BP. Enabling the Pulse Shaper inserts an IrDA-compliant Pulse Shaper between the SDS signals and the SDS pins. Setting bit[0] to a “1” in the IrDA Control Register enables the IrDA block.

A functional block diagram for Pulse Shaper operation is provided in Figure 6.

Table 14. Table Baud Rate Divisor

	19.5 MHz		3.900 MHz	
Baud rate	**Divisor**	**Error**	**Divisor**	**Error**
300 Hz	65000	0.00%	13000	0.00%
450 Hz	43333	0.00%	8667	0.00%
600 Hz	32500	0.00%	6500	0.00%
900 Hz	21667	0.00%	4333	0.01%
1,200 Hz	16250	0.00%	3250	0.00%
1,800 Hz	10833	0.00%	2167	-0.02%
1,900 Hz	10263	0.01%	2053	0.01%
2,400 Hz	8125	0.00%	1625	0.00%
3,600 Hz	5417	0.00%	1083	0.03%
4,800 Hz	4063	0.01%	813	-0.06%
7,200 Hz	2708	-0.01%	542	-0.06%
9,600 Hz	2031	-0.01%	406	0.06%
14,400 Hz	1354	-0.01%	271	-0.06%
19,200 Hz	1015	-0.06%	203	0.06%
28,800 Hz	677	-0.01%	135	0.31%
38,400 Hz	508	0.04%	102	-0.43%
57,600 Hz	339	0.14%	68	-0.43%
76,800 Hz	254	0.04%	51	-0.43%
115,200 Hz	169	-0.16%	34	-0.43%
153,600 Hz	127	0.04%	25	1.56%
230,400 Hz	85	0.43%	17	-0.43%

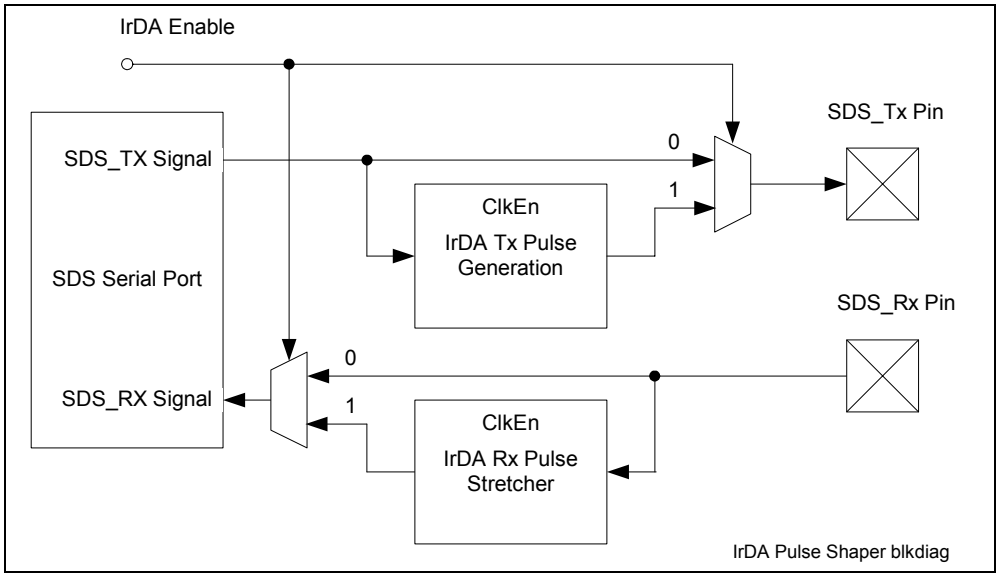


Figure 6. IrDA Pulse Shaper Functional Block Diagram

When the IrDA block is enabled, the following occurs:

1. Using the bit stream output from the SDS Serial Port block on the SDS_RX signal, the IrDA Tx Pulse Shaper generates a short positive pulse for every "0" transmitted by the port and a low level pulse for every "1" transmitted by the port. The duration of pulse is programmable. The output of the Tx Pulse Shaper is routed to the SDS_TX pin.
2. If the IrDA RX Pulse Stretcher recognizes a pulse of the expected duration on the SDS_RX pin, a "0" is generated on the SDS_RX signal for one bit period. If no pulse is detected, a "1" is generated for the duration of the bit period. The expected duration of the received pulse is programmable.

IrDA Registers

There are three IrDA registers:

- IrDA Control Register
- IrDA Baud Register
- IrDA Configuration Register

IrDA Control Register

The IrDA Control Register defines various options on transmit and receive data such as filtering, inversion, loopback and block enable.

IrDA Baud Register

The IrDA Baud Register is used to set the duration of received data in low level input to the SDS block when an IR pulse is detected. The value specified in this register defines the number of 3.9 MHz clocks.

IrDA Configuration Register

The IrDA Configuration Register is used to set the length of the Tx pulse and the expected duration of the Rx pulse. Bits [5:0] set the length of the Tx pulse. The pulse duration is calculated using Equation (1):

$$Pulse\ Duration = (Bit[5:0] + 1) \times System\ \ Clock\ \ Cycle \quad (1)$$

For example, if the bits are set to 0x0F, the pulse duration is seen in Equation (2):

$$Pulse\ Duration = (15 + 1) \times \frac{1}{3.9\ MHz} = 4.1\ \mu sec \quad (2)$$

Bits[13:8] set the expected Rx pulse length. A pulse received on the SDS_RX pin must be of this duration for the Pulse Stretcher to generate a "0" on the SDS_RX pin. The expected pulse duration is calculated the same as the Tx pulse duration. IrDA Register address and default values are specified in Table 6.

Cyclic Redundancy Check Block

The CRC block is used to perform error checking on blocks of data. The ARM Core writes a sequence of byte-wide data to the CRC Data Register, which calculates an output polynomial based on the input sequence. This CRC block supports the standard 16-bit CRC polynomial algorithm as:

$$CRC16 = \left(X^{16} + X^{12} + X^{5} + 1\right)$$

Before writing the data sequence, the output polynomial is reset to 0xFFFF by writing to the CRC Reset Register. The output polynomial can be read at any time and is typically appended to a block of data so that the receiving entity can determine if there were errors in the received data.

Reading from the CRC Data Register gives the output polynomial. The address and default values for the CRC Registers are specified in Table 6.

DMA Controller

The DMA Controller allows blocks of data to be transferred directly between peripherals and memory. The DMA Controller has the following features:

- Full ASB bus master interface
- Six prioritized DMA channels with configurable peripheral request mapping
- Support 11 peripheral request lines
- 2-cycle DMA transfer:
  - ASB read from the Current Source Pointer, storing the data in the Holding Register
  - ASB write to the Current Destination Pointer using the data from Holding Register
- All address pointers are 32-bits wide
- Either, neither, or both Current Source and Current Destination Register can be incremented after each successful DMA transfer
- A configurable Threshold Pointer for each channel can generate interrupt and/or disable the channel when the Incrementing Pointer reaches that address.

When a DMA transfer is required, the ARM sets up the DMA Controller with the block starting address and the block finishing address. If a number of DMA transfers are requested at the same time, the DMA Controller determines which has the highest priority.

There are six DMA channels. The highest priority is Channel 0, and the lowest priority is Channel 5.

During the DMA transfer, the DMA Controller assumes control of the bus and communicates with the SIU to allow the data to be transferred directly between the memory and the peripheral. After each data transfer, the value in the DMA Address Register, which is originally loaded with the block starting address, is incremented. When the block finishing address is reached, the

DMA Controller ceases data transfer after the current transfer has been completed.

DMA Registers

Each DMA channel has the following registers, which are discussed below. DMA Register address and default values are specified in Table 6.

- Status Register
- Control Register
- Two Initial Source Pointer Registers
- Two Initial Destination Pointer Registers
- Two Threshold Pointer Registers
- Two End-of-Buffer Registers
- Two Current Source Pointer Registers
- Two Current Destination Pointer Registers

DMA Status Register

The DMA Status Register reports the operation of the DMA channel. The function of each of the bits in the register is shown in Table 15.

DMA Control Register

The DMA Control Register configures the operation of the DMA channel. The function of each of the bits in the register is shown in Table 16.

Table 15. DMA Status Register Functions

Bit	Function
[15:7]	Reserved
[6]	Channel Enable: 0 = disabled 1 = enabled
[5]	Source Pointer Alignment Error: 0 = normal 1 = error
[4]	Destination Pointer Alignment Error: 0 = normal 1 = error
[3]	Threshold Alignment Error: 0 = normal 1 = error
[2]	End-of-Buffer Alignment Error: 0 = normal 1 = error
[1]	ASB Read Error: 0 = normal 1 = error
[0]	ASB Write Error: 0 = normal 1 = error

Table 16. DMA Control Register Functions

Bit	Function
[15:13]	Reserved
[12:9]	Peripheral Request: 0000 = Gendex_Tx 0001 = Gendex_RX 0010 = DBG_Tx 0011 = SDS_Tx 0100 = DBG_Rx 0101 = SDS_Rx 0110 = Frm_Tx 0111 = Frm_Rx 1000 = SIM_Tx_Rx 1001 = SLOCK 1010 = PREQCLK Others = Channel Disabled
[8:7]	Source Transfer Size: 00 = Byte 01 = Halfword 10 = Word 11 = Reserved
[6:5]	Destination Transfer Size: 00 = Byte 01 = Halfword 10 = Word 11 = Reserved
[4]	Source Pointer Increment Enable: 0 = disabled 1 = enabled
[3]	Destination Pointer Increment Enable: 0 = disabled 1 = enabled
[2]	Threshold Interrupt Enable: 0 = disabled 1 = enabled
[1]	Threshold Stop Enable: 0 = disabled 1 = enabled
[0]	Channel Enable: 0 = disabled 1 = enabled

DMA Initial Source Pointer Register

This pointer register is written to by the ARM to set the starting address of the block to be transmitted. The address written must be aligned with source transfer size as defined in the DMA Control Register. Writing to this register also updates the corresponding Current Source Pointer Register.

DMA Initial Destination Pointer Register

This pointer register is written to by the ARM to set the starting address of the block to be received. The address written must be aligned with destination transfer size as defined in the DMA Control Register. Writing to this register also updates the corresponding Current Destination Pointer Register.

DMA Threshold Register

The ARM writes to this register to set the finishing address at which an interrupt is generated and/or the DMA channel should be disabled. The address written must be aligned with source transfer size as defined in the DMA Control Register. After a DMA transfer is successfully completed in which the address in the incrementing pointer equals the address in the Threshold Pointer:

- The DMA Controller issues the channel's interrupt, if the Threshold Interrupt Enable bit is set.
- Then DMA Controller disables the channel by clearing the enable bit in the DMA Control Register, if the Threshold Stop Enable bit is set.

DMA End-of-Buffer Register

The ARM writes to this register to set the address at which the Current Source and Destination Pointer Registers should be reset. The address written must be aligned with source transfer size as defined in the DMA Control Register. After a DMA transfer is successfully completed in which the address in the incrementing pointer equals the address in the End-of-Buffer Pointer, the Current Source Pointer is reset to the in the Initial Source Pointer. The Current Destination Pointer is reset to the value in the Initial Destination Pointer.

DMA Current Source Pointer Register

The ARM writes to this pointer register to set the starting address of the block to be transmitted. The address written must be aligned with source transfer size as defined in the DMA Control Register. When enabled by Source Pointer Increment Enable bit, this pointer is incremented after each successful DMA transfer until the incrementing pointer equals the End-of-Buffer Pointer.

DMA Current Destination Pointer Register

The ARM writes to this pointer register to set the starting address of the block to be received. The address written must be aligned with destination transfer size as defined in the DMA Control Register. When enabled by Destination Pointer Increment Enable bit, this pointer is incremented after each successful DMA transfer until the incrementing pointer equals the End-of-Buffer Pointer.

Gendex

Gendex is a coprocessor engine for ARM Core to perform the Encryption/Decryption and CRC check/generation on LLC frame when GPRS packet switch data is utilized. It supports both GEA1 and GEA2 algorithm. It will operate in 2 modes. encryption for transmit and decryption for receive. Its operation loading or clock cycles is determined by the frame size and lag time between data request/ready and data sent/received. This engine has the following operation features:

- Inputs
  - Text Input (3-bytes to 1600-bytes) via the ASB I/F
  - 64-bit cipher key, Kc
  - 32-bit initialization vector, Input
  - Direction bit, Dir
  - Frame Length in bytes, Frlen
  - Number of bytes to be CRC'd, NCRC (from the start of the frame)
  - Frame Header Length in bytes, Offset (determines where Ciphering/Deciphering begins in the frame)
- Outputs
  - Text Output (3-bytes to 1600-bytes)

A typical data flow on Gendex block is shown in Figure 7.

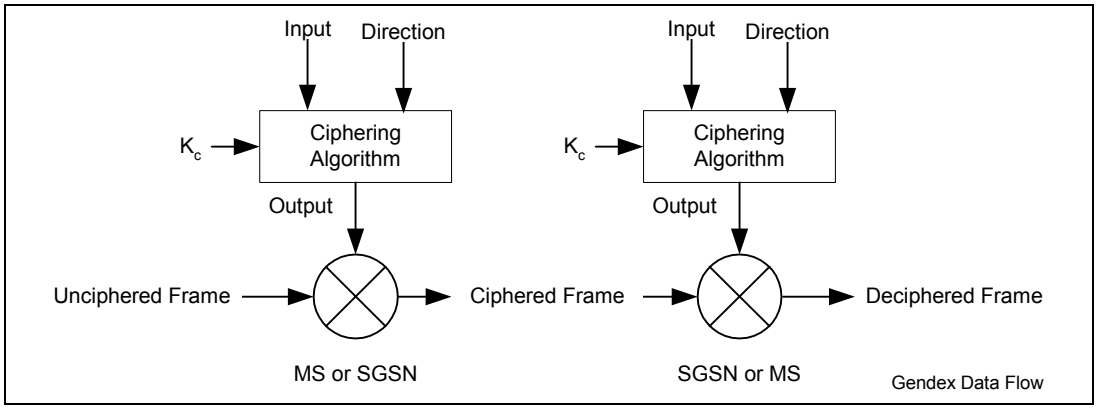


Figure 7. Gendex Data Flow

Gendex Registers

The Gendex block has the following registers, which are discussed below. Gendex Register address and default values are specified in Table 6.

- Gendex Status Register
- Key Input Word Registers
- Input Word Registers
- Gendex Control Register
- Frame Length Register
- CRC Byte Register

Gendex Status Register

The Gendex Status Register reports the operation of the Gendex block. The function of each of the bits in the register is shown in Table 17.

Table 17. Gendex Status Register Functions

Bit	Function
[15:5]	Reserved
[4]	CRC Error: 0 = normal 1 = error
[3]	Write activity: 0 = Write buffer full 1 = waiting for ASB write
[2]	Read activity: 0 = read buffer empty 1 = waiting for ASB read
[1]	Clock request activity: 0 = idle 1 = active
[0]	Gendex activity: 0 = idle 1 = busy

Key Input Word Registers

There are four 16-bit Key Input Registers, which make up the 64-bit key for the encryption/decryption engine.

Input Word Registers

There are two 16-bit Input Registers, which make up the 32-bit initialization vector input data for the encryption/decryption engine.

Gendex Control Register

The Gendex Control Register configures the operation of the Gendex block. The function of each of the bits in the register is shown in Frame Length Register

This register specifies the number of bytes in the Frame. Only bit [10:0] are utilized, bit[15:11] are unused.

CRC Byte Register

This register specifies the number of bytes for CRC calculation/ verification. Only bit[10:0] are used, bit[15:11] are unused.

Table 18.

Frame Length Register

This register specifies the number of bytes in the Frame. Only bit [10:0] are utilized, bit[15:11] are unused.

CRC Byte Register

This register specifies the number of bytes for CRC calculation/ verification. Only bit[10:0] are used, bit[15:11] are unused.

Table 18. Gendex Control Register Functions

Bit	Function
[15:12]	Reserved
[11]	GEA algorithm select: 0 = GEA2 1 = GEA1
[10:9]	Gendex Test Mux Option: 00 = Option 0 01 = Option 1 10 = Option 2 11 = Option 3
[8:7]	Interrupt condition: 00 = interrupt when CRC error has been detected 01 = interrupt when frame processing is complete 10 = interrupt disabled 11 = interrupt disabled
[6]	Data mode select: 0 = 16-bit mode 1 = 32-bit mode
[5]	CRC Enable: 0 = enabled 1 = disabled
[4]	Cipher Enable: 0 = enabled 1 = disabled
[3]	Gendex direction: 0 = Encryption (Tx) 1 = Decryption (Rx)
[2]	Go: 0 = wait 1 = start encipher/decipher
[1]	Gendex block Enable: 0 = disabled 1 = enabled
[0]	Reset: 0 = reset Gendex block 1 = normal

GPIO Ports

The BP provides 27 GPIO signals, which may be used to implement various GSM handset functions. The GPIO signals are divided into five ports:

- Port A with eight signals (GPIO_A[7] to GPIO_A[0])
- Port B with eight signals (GPIO_B[7] to GPIO_B[0])
- Port C with eight signals (GPIO_C[7] to GPIO_C[0])
- Port D with three signals (GPIO_D[2] to GPIO_D[0])

The ports are fully programmable as Inputs/Outputs as follows:

- Each Bit has an Output Enable and an Input Enable.
- Each Bit has an Output Data Register. When Writing, each bit has a Write Enable Mask.
- Each Bit is readable through an Input Data Register.
- Each Bit has a variety of signals, which can be routed out/in through the pad. These optional signals are controlled by both a pair of selection signals that correlate to each GPIO bit, and also to the following set of high level Application-Specific Integrated Circuit (ASIC) control signals (bits in TESTCTL registers. FIELD_TEST, DAIEN, DSPTST, PCMP_MODE, SSP_MODE).

These controls are accessible through GPIO Block registers, except for those listed as TESTBLK controls.

The IO are grouped into the following groups:

- Group0. Ports A and B
- Group1. Port C
- Group2. Port D[2:0]
- Group3. Port D[7:3] (PGA only)

Each Group can adjust the Drive Strength of its member bits via controls in the GPIO Block Registers. Each Group can also have pull-ups/pull-downs adjusted. The ability to pull-up to 10 µA or 100 µA, or pull-down to 10 µA strengths exists for each pad. The values to program are listed in Table 19.

Table 19. Pull-Up/Pull-Down Values to Program

DC Controls	DRIVE Controls
00 = No Pull up/down	000 = 0.8 mA Drive
01 = 10 uA Pull Down	001 = 2.8 mA Drive
10 = 100 uA Pull Up	010 = 4.8 mA Drive
11 = 10 uA Pull Up	011 = 6.8 mA Drive
	100 = 10.8 mA Drive
	101 = 12.8 mA Drive
	110 = 14.8 mA Drive
	111 = 16.8 mA Drive

Port A bit 0 and bit 1 has a permanent 100 uA pull-up.

GPIO Interrupts

Each of the Port B GPIO signals can generate an interrupt to the device Interrupt Controller. These interrupts are routed to the Interrupt Controller before being latched and, therefore, are present even if the system clock signal to the circuit is turned off.

GPIO Registers

The GPIO block has the following registers, which are discussed below. GPIO Register address and default values are specified in Table 6.

- Group 10 Configuration Register
- Group 32 Configuration Register
- Port Control Register
- Port Data Register

For each of the registers, there is a one-to-one mapping between the register bit and the GPIO signal it controls. For example bit[0] in the Port A I/O Select Register enables/ disables the special function of Port A [0].

Group 10 Configuration Register

This register defines the DC and drive control of Group0, Port A and B, and Group1, Port C, Table 20.

Table 20. Group 10 Configuration Register Functions

Bit	Function
[15]	Reserved
[14:12]	Drive control for Group1
[11]	Reserved
[10]	Alternate DC control for Port C 0 = disabled 1 = enabled to set DC control of Port A to override Group 1 setup as: Port C[7:4]. 10 µA pull-down Port C[3]. no pull-up/pull-down Port C[2]. 10 µA pull-up Port C[1]. no pull-up/pull-down Port A[0]. 10 µA pull-up
[9:8]	DC control for Group1
[7]	Reserved
[6:4]	Drive control for Group0
[3]	Alternate DC control for Port B 0 = disabled 1 = enabled to set DC control of Port B to override Group 0 setup as: Port B[7:0]. 10 µA pull-up
[2]	Alternate DC control for Port A 0 = disabled 1 = enabled to set DC control of Port A to override Group 0 setup as: Port A[7:2]. 10 µA pull-down Port A[1:0]. 100 µA pull-up
[1:0]	DC control for Group0

Group 32 Configuration Register

This register defines the DC and drive control of Group 2 Port D[2:0] and Group 3 Port D[7:3], Table 21.

Table 21. Group 32 Configuration Register Functions

Bit	Function
[15]	Reserved
[14:12]	Drive control for Group 3 (PGA only)
[11:10]	Reserved
[9:8]	DC control for Group 3 (PGA only)
[7]	Reserved
[6:4]	Drive control for Group 2
[3:2]	Reserved
[1:0]	DC control for Group 2

Port Control Register

Each GPIO signal has a companion configuration bit that selects the GPIO signal as an input or an output.

- If the bit is set to "0", the signal is configured as an output.
- If the bit is set to "1", the signal is configured as an input as shown in Table 22.

Table 22. Port Control Register Functions

Bit	Function
[15:8]	Output Enable 0 = enabled 1 = tri-state output
[7:0]	Input Enable 0 = disabled 1 = enabled

Port Data Register

Each of the ports has a Port Data I/O Register. The ARM writes to this register to output data on the port pins and reads from this register to read data input on the port pins as shown in Table 23.

Table 23. Port Data Register Functions

Bit	Function
[15:8] Write	Output Data 0 = low level on output data register 1 = high level on output data register
[7:0] Write	Output Enable 0 = no change on output data pins 1 = change output data pins corresponding to [15:8] bits
[15:8] Read	Output Data 0 = low level on output data register 1 = high level on output data register
[7:0] Read	Input Data 0 = low level on input data pin 1 = high level on input data pin

Each of the GPIO signals has an associated I/O select bit. Several of the GPIO signals are multiplexed with other special function signals. The alternate function is enabled when a "1" is written to the associated I/O select bit. These optional signals are controlled by both a pair of selection signals with correlation to each GPIO bit, and a set of high level ASIC control signals (bits in TESTCTL registers. TESTP_B, FIELD_TEST, DAIEN, DSPTST).

For example, GPIO Port C[4] is multiplexed with the output from the BP PWM circuit:

- If a "1" is written to the Port C[4] I/O select bit, the device PWM circuit output is routed to the GPIO pin.
- If a "0" is written to the Port C[4] I/O select bit, the GPIO pin is configured as a GPIO signal.

The alternate function associated with each of the GPIO signals is shown in Table 24.

GPIO initial conditions are shown in Table 25.

Table 24. GPIO Alternate Functions

IOSEL = 0	IOSEL = 1	DAIEN = 1	FldTest = 1	IOSEL = 0	IOSEL = 1	DAIEN = 1	FldTest = 1
GPIO Port A 7	Timer2 Pulse	default	default	GPIO Port C 7	ECLK	default	DSP_GP0
GPIO Port A 6	DSP CLK	default	default	GPIO Port C 6	Debug Rx	DAI_Rx	Default
GPIO Port A 5	DSP NIRQ	default	default	GPIO Port C 5	Debug Tx	DAI_Tx	default
GPIO Port A 4	NFIQ	default	default	GPIO Port C 4	PWM Out	default	default
GPIO Port A 3	BCLK	default	default	GPIO Port C 3	SIM Reset	default	default
GPIO Port A 2	Rx Byte Strobe	default	DSP_GP1	GPIO Port C 2	EBI Chip Select 6	default	default
GPIO Port A1	KYRTN(4)	default	default	GPIO Port C 1	SIM VCC Enable	default	default
GPIO Port A 0	GPIO Port A 0	default	default	GPIO Port C 0	EBI Chip Select 5	default	default
GPIO Port B 7	Rx Data 7	default	default				
GPIO Port B 6	Rx Data 6	DAI_RST	default	GPIO Port D 2	EBI Chip Select 4	default	DSP_Tx
GPIO Port B 5	Rx Data 5	default	default	GPIO Port D 1	EBI Chip Select 3	default	default
GPIO Port B 4	Rx Data 4	default	default	GPIO Port D 0	EBI Chip Select 2	default	DSP_Rx
GPIO Port B 3	Rx Data 3	DAI_CLK	default				
GPIO Port B 2	Rx Data 2	default	default				
GPIO Port B 1	Rx Data 1	default	default				
GPIO Port B 0	Rx Data 0	default	default				

Table 25. GPIO Initial Conditions

IOSEL = 0	IOSEL	OE	IE	DC	IOSEL = 0	IOSEL	OE	IE	DC
GPIO Port A 7	0	off	on	down 10	GPIO Port C 7	0	off	on	down 10
GPIO Port A 6	0	off	on	down 10	GPIO Port C 6	0	off	on	down 10
GPIO Port A 5	0	off	on	down 10	GPIO Port C 5	0	off	on	down 10
GPIO Port A 4	0	off	on	down 10	GPIO Port C 4	0	off	on	down 10
GPIO Port A 3	0	off	on	down 10	GPIO Port C 3	1	on	off	off
GPIO Port A 2	0	off	on	down 10	GPIO Port C 2	0	off	on	up 10
GPIO Port A1	0	off	on	up 100	GPIO Port C 1	1	on	off	off
GPIO Port A 0	0	off	on	up 100	GPIO Port C 0	0	off	on	up 10
GPIO Port B 7	0	off	on	up 10					
GPIO Port B 6	0	off	on	up 10	GPIO Port D 2	0	off	on	up 10
GPIO Port B 5	0	off	on	up 10	GPIO Port D 1	0	off	on	up 10
GPIO Port B 4	0	off	on	up 10	GPIO Port D 0	0	off	on	up 10
GPIO Port B 3	0	off	on	up 10					
GPIO Port B 2	0	off	on	up 10					
GPIO Port B 1	0	off	on	up 10					
GPIO Port B 0	0	off	on	up 10					

Keypad Interface

The keypad interface consists of eight strobe lines and four return lines. This allows for a keypad matrix of up to 32 keys. The strobe lines are GPIO-type signals, and the return lines are dedicated signals. Unused strobe lines can be used as extra GPIO signals if required. If a keypad matrix of more than 32 keys is required, one of the GPIO signals, GPIO Port A[1], can be configured as a return line to give a total matrix of 40 keys. The GPIO is configured as a keypad return line by setting the Port A[1] I/O select bit to “1”. See the GPIO section for more information).

Figure 8 illustrates a typical keypad interface circuit. The strobe lines are output from the BP and connected to the keypad columns. The return lines are input to the BP and connected to the keypad rows.

Keypad Scan

Once every frame, that is, every 4.6 ms, the BP drives one of the strobe lines low and all of the others high. If a key connected to the strobe line that is driven low is pressed during this time, the return line that is also connected to this key is pulled low. This triggers an interrupt to the microcontroller to indicate that a key has been pressed. Since the microcontroller knows which strobe line it is driving low and which return line has been driven low by the keypress, it can uniquely identify the key pressed.

A complete scan of the entire keypad requires 4.6 ms multiplied by the number of strobe lines, see Equation (3).

$$Complete\ \ keyboard\ \ scan = 4.6\ \ ms \times 8 = 36.8\ \ ms \qquad (3)$$

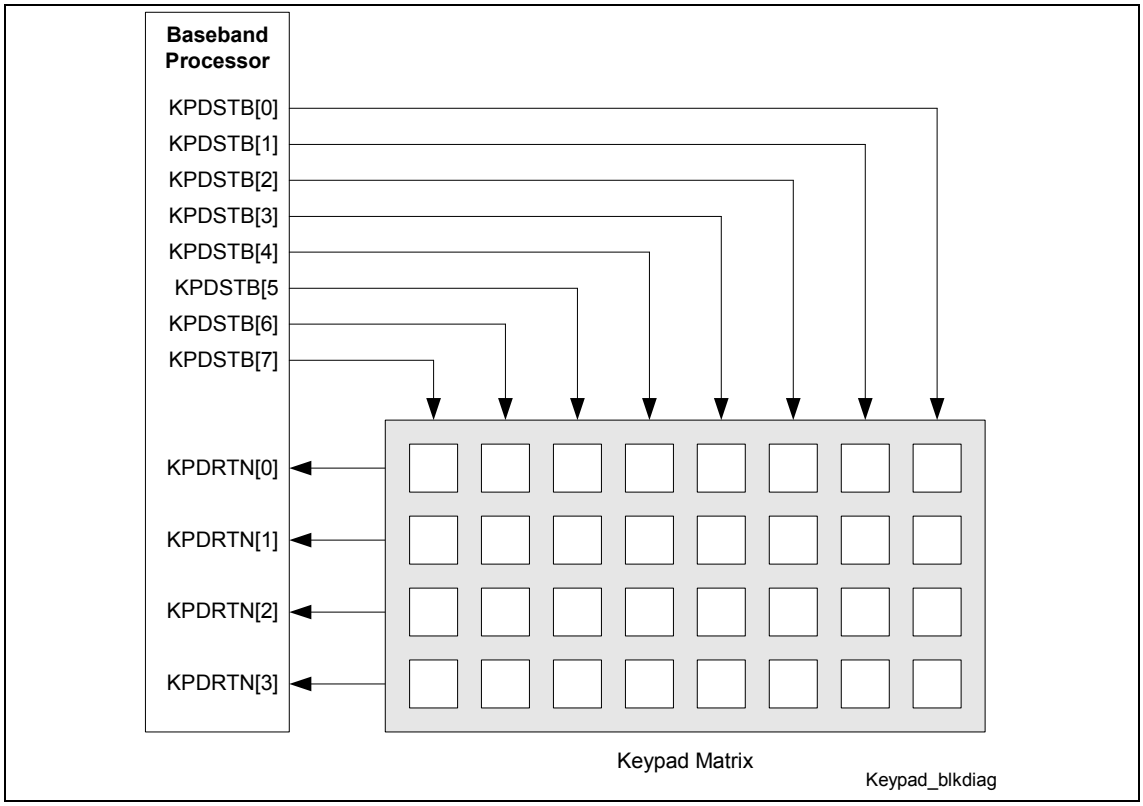


Figure 8. Typical Keypad Interface Circuit

Keypad Registers

The following keypad registers are used to configure and monitor the keypad interface. The address and default values for the keypad registers are specified in Table 6.

- Keypad I/O Configuration Register
- Keypad Strobe Register
- Keypad Return Lines Register

Keypad I/O Configuration Register

The Keypad I/O Configuration Register is used to set up the strobe lines as either inputs or outputs. Each of the strobe lines has a related bit that sets the direction of the pin. There is a one-to-one mapping between the bits and the keypad strobes, that is, bit 0] configures keypad strobe 0, bit[1] configures keypad strobe 1.

- If the bit is set to “0”, the pin is configured as an output.
- If the bit is set to “1”, the pin is configured as an input.

The register default value is 0xFF, which configures all the keypad strobes as BP outputs.

Keypad Strobe Register

The Keypad Strobe Register is used to specify the output level of the strobe lines. Data written to this register appears on the strobe lines, assuming that the Configuration Register has been written to set up the signals as outputs. There is a one-to-one mapping between the bits and the keypad strobes:

- Bit[0] drives keypad strobe 0.
- Bit[1] drives keypad strobe 1.

Keypad Return Lines Register

The Keypad Return Lines Register is used to read the state of each of the keypad return lines. Keypad Return 4 is multiplexed with GPIO Port A[0], so the related GPIO I/O select bit must be set to configure the pin as a keypad return signal.

There is a one-to-one mapping between the bits and the keypad return lines:

- Bit[0] stores the state of keypad return line 0.
- Bit[1] stores the state of keypad return line [1].

Annunciator

The BP annunciator circuitry generates a signal to drive a handset buzzer. Some external circuitry may be required to drive the buzzer depending on its electrical characteristics.

The annunciator circuitry output is multiplexed with GPIO Port C[4]. To select the PWM output to the pin, the I/O select bit for GPIO Port C[4] must be set to “1”, see the GPIO Ports section for more information.

There are two identical but separately programmable PWM circuits on the BP. The outputs from the two circuits are toggled at a rate of 1.95 MHz to generate the annunciator output from the device. The annunciator output circuitry and the buzzer driving circuitry are shown in Figure 9.

PWM Operation

The basic operation of each of the PWM circuits is described below.

The PWM uses a reference clock of 1.95 MHz and 3.9 MHz to generate two pulse trains:

- A low frequency pulse train programmable from 200 to 500 Hz. This is the Tone signal.
- A high frequency pulse train at 39 kHz with a variable duty cycle. This is the Mod signal.

The two pulse trains are mixed and filtered to produce the PWM output. The frequency of this output is the frequency of Tone; the volume is dependent on the duty cycle of Mod. The frequency of Tone and the duty cycle of Mod are programmable so that complex annunciator tones can be generated.

A PWM functional block diagram is shown in Figure 10.

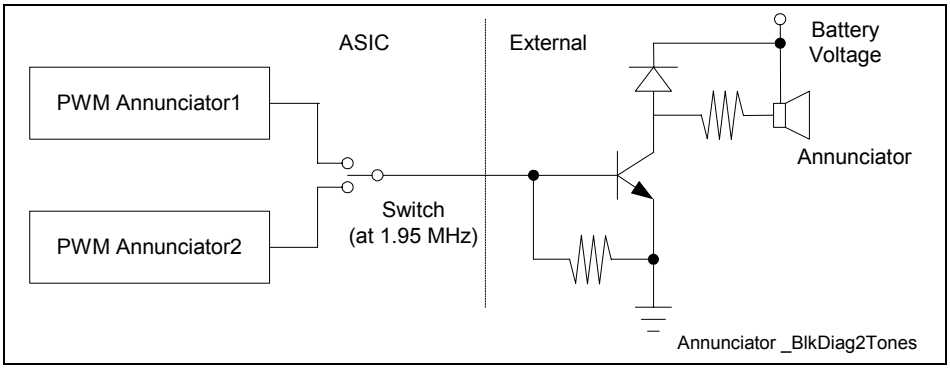


Figure 9. Annunciator Output Circuitry and Buzzer Driving Circuitry

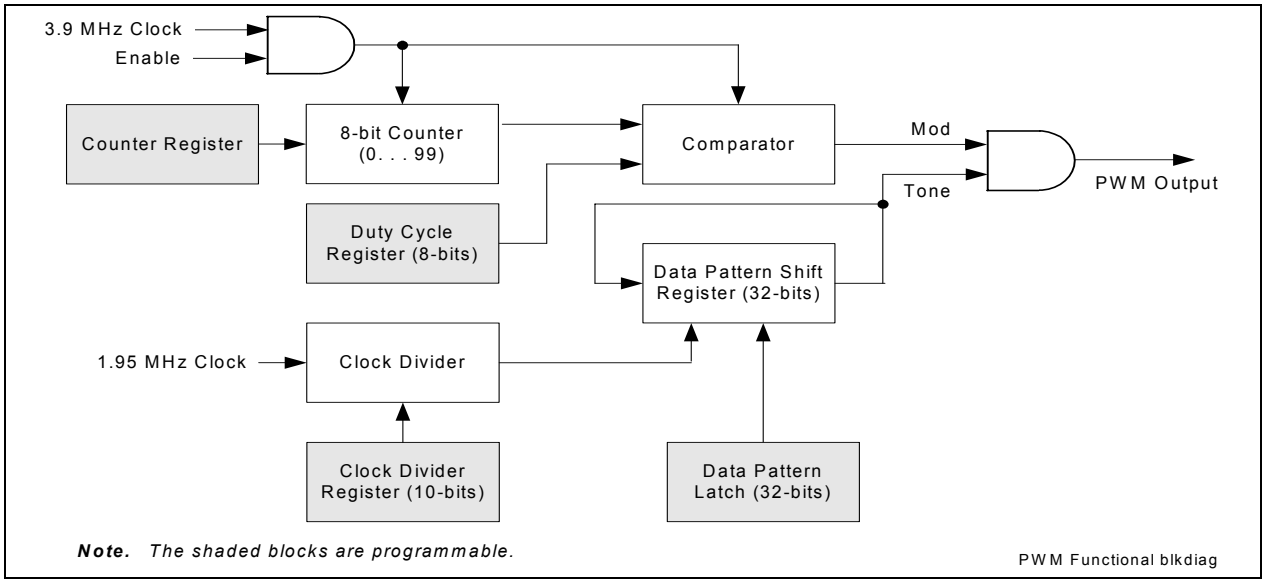


Figure 10. PWM Functional Block Diagram

The counter circuit is an 8-bit counter that is clocked at 3.9 MHz. The counter counts to the value set in the Counter Register and is then reset to “0”. The Counter Register default value is decimal 100.The Duty Cycle Register is a programmable 8-bit

register. The duty cycle for Mod is written to this register. The comparator function compares the counter output with the value of the Duty Cycle Register. The comparator output is determined by the following rule:

If Counter Value ≤ Duty Cycle Register, Comparator Output = 1, Else Comparator Output = 0

The comparator output is the Mod signal. Writing to the Duty Cycle Register and to the Counter Register programs the Mod signal duty cycle. If the Counter Register default value is used, decimal 100, the Mod signal frequency is 39 kHz (3.9 MHz/100).

An example of Mod signal generation is shown in Figure 11. In this example, the Counter Register default value of 100 is used.

The Clock Divider Register is a programmable, 10-bit register. The divider circuit divides down the 1.95 MHz input to the circuit by the register value to yield an output signal with frequency F. The divider output is used to clock the Data Pattern Shift Register.

The Data Pattern Shift Register is a 32-bit serial shift register. The programmable Data Pattern Latch Register contents are downloaded to the Data Pattern Shift Register to set the shift register initial value.

The shift register output is the tone pulse train, Tone. This output is also fed back to the register input so that the same series of 32 bits are continuously cycled through the register. The data pattern is cycled at a frequency of f2/32.

A Tone signal generation example is shown in Table 26.

The Mod and Tone pulse trains are ANDed to generate the annunciator circuit output.

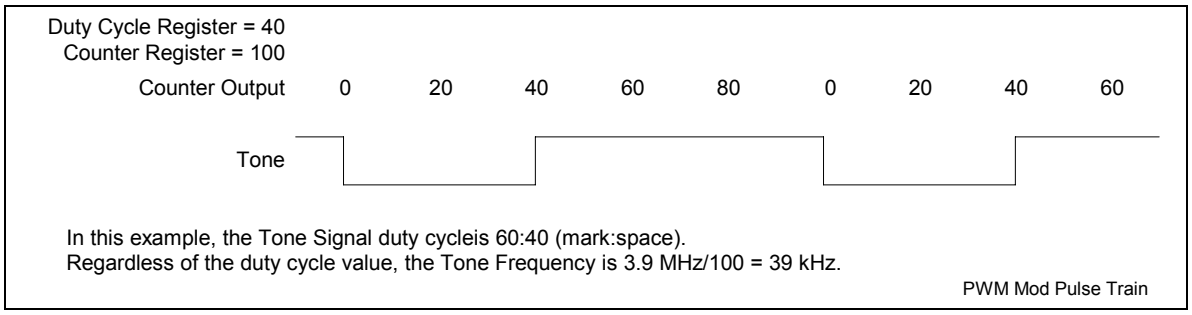


Figure 11. Generation of PWM Mod Pulse Train

Table 26. Example Generation of the Tone Signal

Definition	Value
Programmable Parameters	
Clock Divider Register	250
Data Pattern Shift Register	0xF0F0F0F0F0F0F0 (11110000111100001111000011110000b)
Tone Frequency Calculation	
Clock Divider Input Clock	1.95 MHz
Clock Divider Output	1.95 MHz/250 = 7800 Hz (F)
Data Pattern Frequency	F/32 = 7800 Hz/32 = 243.75 Hz
With the above data pattern, there are four signal cycles per data pattern cycle.	
Therefore, the Tone frequency is:	243.75 Hz × 4 = 975 Hz
and the Tone duty cycle	50%

PWM Registers

Table 27 shows the address and function of each of the PWM registers. The default value of each of the registers is specified in Table 6.

Autobaud

The Autobaud circuit monitors the SDS_RX input and automatically detects the baud rate of the data received, whether the data is formatted into seven or eight-bit words and whether the parity is odd or even. There are certain restrictions on the baud rate, word length, and characters that can be detected.

When the Autobaud block is enabled, the ARM microcontroller configures all the SDS port parameters except for the baud rate, word length, and parity, which are provided by the Autobaud circuit.

After successful derivation of these parameters, the Autobaud circuit programs the SDS port with the information, enables the

SDS port, and disables itself. The SDS port starts to receive characters immediately after the “t”, “T”, or “/” is received as the second character. While the Autobaud block is reprogramming the SDS port, the ARM cannot access the port configuration registers.

After the Autobaud block disables itself, an interrupt is sent to the ARM to indicate that the SDS port is receiving valid data. The values of the characters that were decoded are stored so the ARM can determine which characters were detected.

Autobaud Limitations

The Autobaud circuit can detect the following limitations, which are described below:

- Baud rates
- Character combinations
- Data formats

Table 27. PWM Registers

Register Address (Hex)	Register Function
0x0040A00	Control Register for PWM circuit 1: Bit[15:2]. Reserved. Bit[1]. Read the value of the 32-BitShift Register (0 = read latch, 1 = read Shift Register). Bit[0]. PWM circuit 1 enable (0 = off, 1 = on).
0x0040A10	Control Register for PWM circuit 2: Bit[15:2]. Reserved. Bit[1]. Read the value of the 32-BitShift Register (0 = read latch, 1 = read Shift Register). Bit[0]. PWM circuit 2 enable (0 = off, 1 = on).
0x0040A02	Annunciator Duty Cycle Register for PWM circuit 1: Bit[15:8]. Reserved. Bit[7:0]. Duty cycle.
0x0040A12	Annunciator Duty Cycle Register for PWM circuit 2: Bit[15:8]. Reserved. Bit[7:0]. Duty cycle.
0x0040A04	Divider Register for PWM circuit 1: Bit[15:10]. Reserved. Bit[9:0]. Divider ratio.
0x0040A14	Divider Register for PWM circuit 2: Bit[15:10]. Reserved. Bit[9:0]. Divider ratio.
0x0040A06	Counter Register for PWM circuit 1: Bit[7:0]. Counter value.
0x0040A16	Counter Register for PWM circuit 2: Bit[7:0]. Counter value.
0x0040A08	Data pattern latch for PWM circuit 1: Bit[31:0]. 32-Bitdata pattern.
0x0040A18	Data pattern latch for PWM circuit 2: Bit[31:0]. 32-Bitdata pattern.

Baud Rates. The following baud rates can be detected:

- 230.4 kbps
- 115.2 kbps
- 57.6 kbps
- 38.4 kbps
- 19.2 kbps
- 9.6 kbps
- 4.8 kbps
- 2.4 kbps

When a low edge is detected on the SDS_RX signal, which indicates a start bit, the Autobaud circuit:

- Starts to search for the required set of start characters
- Determines the baud rate
- Determines whether the data is formatted into 7-bit or 8-bit words
- Detects if parity is odd or even

Character Combinations. The following character combinations can be detected:

- at
- AT
- a/
- A/

The detection circuit allows for successive “a” or “A” characters to be received. For example, “aat” and “aAT” are valid combinations.

Data Formats. The following data formats can be detected:

- 7-bit data, no parity, 2 stop bits
- 7-bit data, even parity, 1 stop bit
- 7-bit data, odd parity, 1 stop bit
- 8-bit data, no parity, 1 stop bit

Autobaud Registers

The Autobaud block has the following registers, which are used to configure and monitor the Autobaud function. The registers are discussed below. Autobaud Register address and default values are specified in Table 6.

- Autobaud Control Register
- Autobaud Status Register
- Failure Register
- Character Register
- Temporary Character Register
- Decision Value Registers
- Baud Count Register

Autobaud Control Register

The Autobaud Control Register is used to control the operation of the Autobaud function. The function of each of the bits in this register is described in Table 28.

Table 28. Autobaud Control Register Functions

Bit	Signal	Function
[15:8]	N/A	Reserved
[7]	Reset Autobaud	Force a soft-reset Autobaud block. 0 = off 1 = on
[6]	SDS Control enable	Enable Autobaud block to override SDS configuration. 0 = off 1 = on
[5]	Auto mode enable	Enable Autobaud block to switch from transmit mode to receive mode automatically. 0 = off 1 = on
[4]	SDS port loopback disable	Indicate whether or not the Autobaud circuit will attempt to disable the hardware loopback in the SDS port on successful detection of an AT sequence. 0 = no control on hardware loopback 1 = disable of hardware loopback
[3]	Select test input	Select the input to monitor 0 = SDS_Rx 1 = TST_Din
[2]	Reuse past protocol enable	Enable Word length, Stop bit and Parity protocol to be reused from previous detection. 0 = off 1 = on
[1]	Failure interrupt enable	Enable interrupt to the ARM if a failure is detected. 0 = off 1 = on
[0]	Autobaud enable	Enable the Autobaud function: 0 = off 1 = on

Autobaud Status Register

The Autobaud Status Register is used to provide the current status of the Autobaud function. The function of each of the bits in this register is described in Table 29. This register is read only.

Table 29. Autobaud Status Register Functions

Bit	Signal	Function
[15:8]	N/A	Reserved
[7]	Stop bit detected	1 = 2 stop bit detected 0 = 1 stop bit detected
[6]	Active	1 = start bit detected and character sequence detection in progress. 0 = no activity
[5]	Correct detection	0 = start bit detected 1 = Autobaud detection completed successfully
[4:3]	Parity	00 = parity detected as no parity 01 = parity detected as even parity 10 = parity detected as odd parity 11 = Invalid
[2]	Word length	0 = word length successfully detected as seven bits 1 = word length successfully detected as eight bits
[1]	Second character detected	0 = start bit falling edge detected 1 = second character successfully compared with the set of possible characters (“t,” “T,” “/”)
[0]	First character detected	0 = start bit falling edge detected 1 = first character successfully compared with the set of possible characters (“a,” “A,” etc.)

Failure Register

The Failure Register is used to detect Autobaud failures. If the Autobaud detection fails, one of the Failure Register bits is set to “1.” The bit that is set depends on the failure mode. Bits[8:0] in this register are cleared to “0” when the start bit is detected. If the Control Register Failure Interrupt Enable bit is set to “1,” an interrupt to the ARM is generated when any of these bits is set. Register bit functions are described in Table 30.

Table 30. Failure Register Functions

Bit	Failure Mode
[15:9]	Reserved
[8]	No previous word/parity
[7]	“At/aT” error
[6]	Second character stop bit error
[5]	Second character match error
[4]	Second character start bit error
[3]	First character stop bit error
[2]	First character match error
[1]	Baud counter overflow
[0]	Start bit too short

Character Register

Bit[7:0] store the first character detected during the last Autobaud attempt. Bit[15:8] store the second character detected during the last Autobaud attempt.

Temporary Character Register

When the Autobaud circuit processes an incoming character, the character bits are stored in bit[9:0] of this register. If the Autobaud function fails, the actual bits received can be read from this register.

Decision Value Registers

The Decision Value Registers are used to determine the baud rate of the incoming data stream by counting the number of 3.9 MHz clock cycles in the start bit. The start bit is a “0” and the first bit of the “A” or “a” character is a “1”.

Therefore, start bit duration can be measured by counting the number of clock cycles for which the SDS_RX signal is low. Based on the number of clock cycles counted, the Autobaud circuit decides the baud rate value, and programs the SDS baud rate accordingly.

There are eight Decision Value Registers, which are programmed to set the lower and upper cycle count thresholds for eight different baud rates. There is a threshold overlap in that the upper cycle count threshold for one baud rate is also the lower cycle count threshold for the next higher baud rate.

When the Autobaud circuit has counted the number of 3.9 MHz clock cycles in the start bit, it then compares this number of counts with values in the Decision Value Registers. If it detects that the number of counts lies between two threshold values set by the registers, then the contents of the corresponding Timer Value are written to the SDS Port Divider Registers to set the port transmit and receive baud rate.

For the Decision Value Registers, bits[10:0] set the count thresholds. Bits[15:11] are unused.

Baud Count Register

Bit[10:0] store the number of 3.9 MHz cycles that the Autobaud circuit has counted in the start bit received on the SDS Rx line.

Escape Sequence Detection and Flow Control

The function of these blocks is to detect an escape sequence or the XON/XOFF flow control characters on the SDS serial port. The escape sequence is a programmable series of characters, normally “+++”, used in some serial communication protocols. The escape sequence is separated from other data by a large guard time.

The detection functions monitor the data bytes received on the SDS_RX line. If the escape character sequence, the XON

character, or the XOFF character is detected, the associated flag is set.

Registers for Flow Control/Escape Sequence Detection

The Flow Control/Escape Sequence Detection registers include the following:

- Control Register
- Status Register
- Escape Sequence Character Register
- XON Character Register
- XOFF Character Register
- Escape Sequence Timout Register

The address and default values for the Escape Sequence Detection and Flow Control Registers are specified in Table 6.

Flow Control/Escape Sequence Detection Control Register

The Flow Control/Escape Sequence Detection Control Register is used to configure the operation of the Escape Sequence Detection and Flow Control circuitry. The function of each of the bits in this register is described in Table 31.

Table 31. Flow Control/Escape Sequence Detection Control Register Functions

Bit	Signal	Function
[15:4]	N/A	Reserved
[3]	Counter enable	Enable/disable the guard time counter used during the escape sequence detection: 0 = counter disabled 1 = counter enabled
[2]	XOFF enabled	0 = XOFF detection disabled 1 = XOFF detection enabled
[1]	XON enabled	0 = XON detection disabled 1 = XON detection enabled
[0]	Escape sequence enable	0 = escape sequence detection disabled 1 = escape sequence detection enabled

Flow Control/Escape Sequence Detection Status Register

The Flow Control/Escape Sequence Detection Status Register is used to indicate the detection status for each of the character sequences, for example, escape sequence, XON, XOFF. The function of each of the register bits is described in Table 32.

Table 32. Flow Control/Escape Sequence Detection Status Register Functions

Bit	Signal	Function
[15:3]	N/A	Reserved
[2]	XOFF detected	Set to "1" if the XOFF character has been detected. The ARM can clear this bit.
[1]	XON detected	Set to "1" if the XON character has been detected. The ARM can clear this bit.
[0]	Escape sequence detected	Set to "1" if the escape sequence has been detected. The ARM can clear this bit.

Escape Sequence Character Register

The Escape Sequence Character Register is used to store the escape sequence character. Bit[7:0] hold the 8-bit character to be compared with the SDS_RX input data to determine if the escape sequence has occurred. The default character is "+". The character must occur three times, for example, "+++", for the escape sequence to be detected.

XON Character Register

The XON Character Register is used to store the XON character. Bit[7:0] store the 8-bit character that is to be compared with the SDS_Rx input data to determine if the XON character has been received.

XOFF Character Register

The XOFF Character Register is used to store the XOFF character. Bit[7:0] store the 8-bit character that is to be compared with the SDS Rx input data to determine if the XOFF character has been received.

Escape Sequence Timeout Register

The Escape Sequence Timeout Register is used to specify the guard band time. The escape character sequence has a guard band before and after the escape sequence characters. Bit[7:0] of this register set the guard band. The guard band is set as the register contents multiplied by 20 ms. The register value should be between 1 and 255.

Bit[15:8] of the Escape Sequence Timeout Register are unused.

Real Time Clock

The RTC provides the handset with the current time in seconds, minutes, hours, days, months, and years. The RTC maintains the time information under primary power or battery backup conditions. The RTC provides real-time information that can be displayed on the handset Liquid Crystal Display (LCD) or used to calculate call duration.

The RTC also has a programmable alarm, which can power down or power up the handset besides providing a time-related

alarm to the user. The RTC uses an external 32.768 kHz crystal as a timing reference.

RTC Supply

The RTC has a dedicated supply input pin, VRTC. Typically, the primary source for the circuit is the handset battery, so that the RTC can continue to keep track of real time even when the handset is powered off. A lithium cell can be used as a secondary power source, so that the RTC remains powered when the handset battery is removed.

The RTC operates over a voltage range of 2.3 V to 2.6 V. The typical current consumption of the RTC is 200 μA.

RTC Crystal

The RTC uses a 32.768 kHz crystal as a timing reference.

RTC Registers

The RTC block includes the following registers:

- RTC Control Register
- Time Interval Registers
- Alarm Registers

The address and default values for the RTC Registers are specified in Table 6.

RTC Control Register

The RTC Control Register is used to reset the RTC. Writing any data to the register resets all the time and alarm registers. Data written to this register, or read from this register, is “do not care.”

The Most Significant Bit (MSB) of each of the time interval registers, for example, seconds, minutes, hours, is a “busy” bit. If set, it indicates that a one second edge has occurred, and the registers are in the process of being updated. Reading from the RTC Control Register resets the “busy” bit in each of the registers.

Time Interval Registers

The Time Interval registers contain the RTC time values. Six individual registers separately track the time in seconds, minutes, hours, days, months, and years. Each register is 8- bits wide. The MSB of each register, bit[7], is a “busy” bit that indicates when a one second edge has occurred and the registers are in the process of being updated. The remaining bits, bit[6:0], indicate the time.

Reading the contents of each of these registers indicates the current time. Writing to any of the registers increments the value of the register by 1. For example, writing to the months register increments the month by 1. The data written to the register is “do not care.”

Alarm Registers

The Alarm registers are used to set the RTC alarm time. When the time indicated by the Time Interval Registers matches the time set in the Alarm Registers, the alarm output from the RTC circuit becomes active. The alarm signal is output from the BP, so it can be used to activate or deactivate other parts of the system.

There are separate registers for seconds, minutes, hours, days, months and years. Writing to these registers sets the time at which the next alarm occurs. Reading from the registers returns the time currently programmed for the alarm.

Skyworks Serial Bus

The Skyworks serial bus is an asynchronous, full duplex serial interface that the BP uses to communicate with other handset system components. The Skyworks serial bus is used for intra-device communications. The BP acts as bus master and the external component is the bus slave. The maximum data transfer rate is 100 kbits/sec.

The bus uses the following signals, are bi-directional and have open drain outputs. The voltages for input and output logic high and logic low levels are specified in Table 33.

- SRLDATA
- SRLCLK

Figure 12 and Table 34 show the timing for data transfer using the Skyworks serial bus.

Table 33. Serial Bus Electrical Characteristics

Parameter	Symbol	Test Conditions	Min	Typical	Max	Units
Input High voltage	V_{IH}		1.5			V
Input low voltage	V_{IL}				0.5	V
Output high voltage	V_{OH}	@ 500 µA I_{OH} (Note 1)	2.4			V
Output low voltage	V_{OL}	@ 3.2 mA I_{OL} (Note 2)			0.4	V
Note 1: *I_{OH} is the maximum source current for a "1" output.* ***Note 2***:: *I_{OL} is the maximum sink current for a "0" output.*						

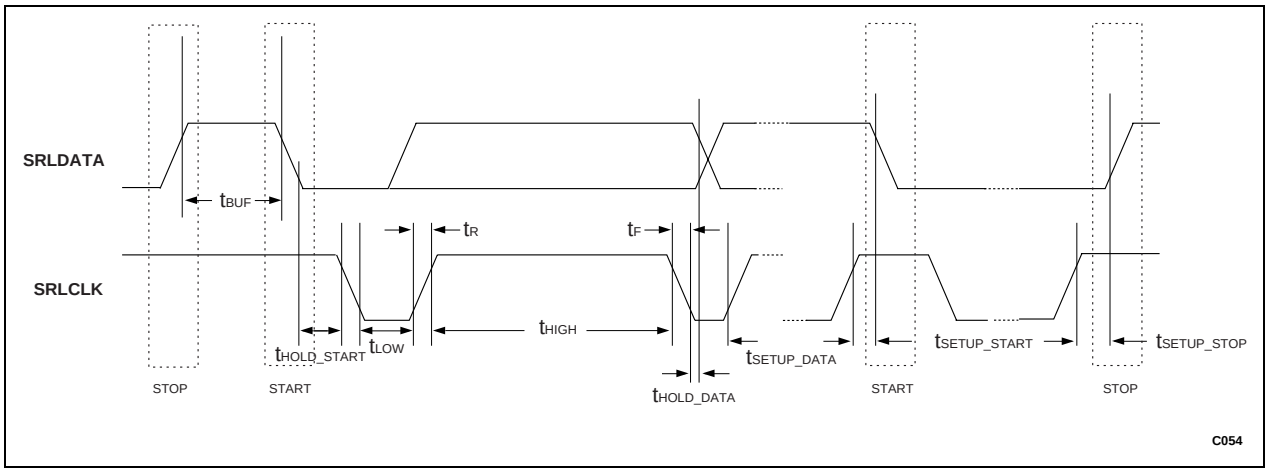


Figure 12. Skyworks Serial Bus Timing Data Diagram

Table 34. Skyworks Serial Bus BP Timing Requirements

Parameter	Symbol	Min	Max	Units
Serial clock frequency	f_{SERIAL_CLOCK}		100	kHz
Time the bus must be free before a new transmission can start	t_{BUF}	4.7		µs
Hold time start condition. After this period the first clock pulse is generated	t_{HOLD_START}	4		µs
Low period of the clock	t_{LOW}	4.7		µs
High period of the clock	t_{HIGH}	4		µs
Setup time for start condition (only relevant for a repeated start condition)	t_{SETUP_START}	4.7		µs
Hold time SRLDATA	t_{HOLD_DATA}	0 **(Note 1)**		µs
Setup time SRLDATA	t_{SETUP_DATA}	250		ns
Rise time of both SRLDATA and SRLCLK lines	t_R		1	µs
Fall time of both SRLDATA and SRLCLK lines	t_F		300	ns
Setup time for stop condition	t_{SETUP_STOP}	4.7		µs

Note 1. *A transmitter must internally provide a hold time to bridge the undefined region (300 ns maximum) of the falling edge of SRLCLK. All values referenced to V_{IH} and V_{IL} levels.*

Data Transfer Protocol

The data on the SRLDATA line must be stable while the SRLCLK signal is high. When the SRLCLK signal is low, the SRLDATA line can change level.

There are exceptions to this rule. When a high-to-low transition occurs on the SRLDATA line while SRLCLK is high, a start condition is indicated, that is, the start of the transfer of an undefined number of bytes over the bus. The bus protocol only allows for transfers of byte-wide data. A low-to-high transition on the data line while the clock line is high indicates a stop condition, that is, the end of the transfer of one or more bytes of data over the bus.

The bus master, that is, the BP, always generates the start and stop conditions. The bus is considered to be busy after the start condition and is free again after the stop condition. Figure 13 shows the bit formatting for data transfers over the bus.

The bus only allows for byte-wide data transfers. An unrestricted number of bits can be transferred between each start and stop condition. The MSB is transmitted first.

After each byte transfer, the bus master generates an acknowledge-related clock pulse. During this clock pulse, the party that received the data must acknowledge that the data has been received by pulling the SRLDATA line low. The transmitter releases the SRLDATA line during this clock period to allow the receiver to drive the line.

All data transactions between bus master and the bus slave are initiated when the BP issues the Serial Bus Start condition. The BP then sends the device address over the SRLDATA line. The device address consists of seven bits of address and one bit that specifies whether the operation is a read or write (from the perspective of the bus master).

After the device address, the bus master sends the register address. Following the register address, either the BP, write operation, or the bus slave, read operation, places the data on the bus by. If more than one byte of data is being transferred, the register address is the starting address and subsequent bytes are read from or written to sequential registers.

Table 14 through Figure 17 shows the sequence of events for single or multiple byte read and write operations over the Skyworks serial bus.

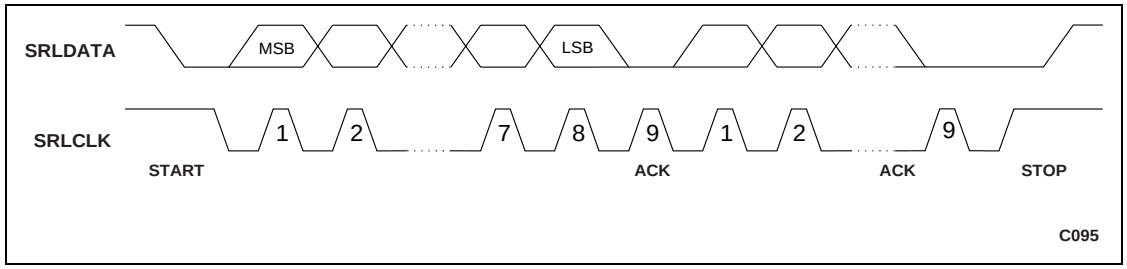


Figure 13. Serial Interface Data Format Diagram

Figure 14. Single Byte Write Operation

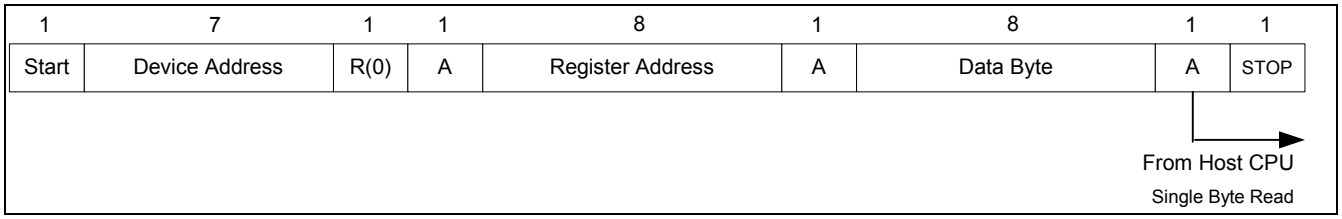


Figure 15. Single Byte Read Operation

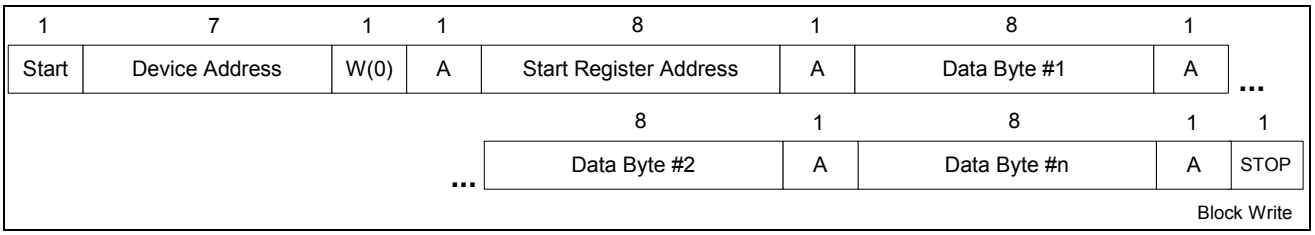


Figure 16. Sequential or Block Write Operation

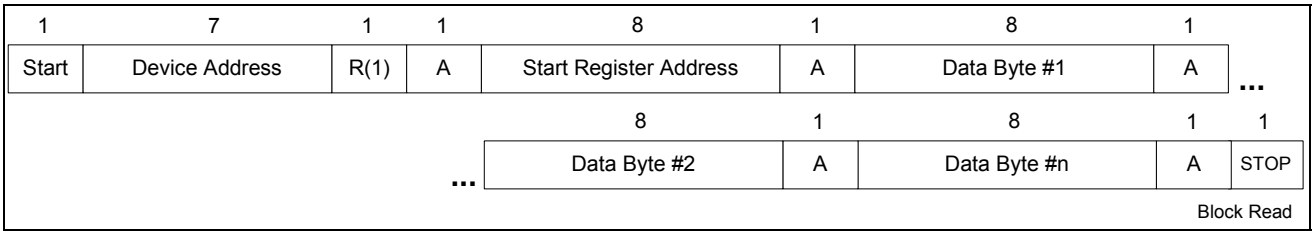


Figure 17. Sequential or Block Read Operation

Serial Bus Registers

The following registers, which are described below, control the operation of the Skyworks serial bus data and clock signals. The address and default values for the Serial Bus Registers are specified in Table 6.

- Serial Data Register
- Serial Clock Register

Serial Data Register

The Serial Data Register is used to specify the logic level of the serial data output. Writing to bit[0] of this register places the written logic level on the serial data pin. Reading bit[1] of the

register returns the actual logic level of the serial data pin. All other bits in this register are unused.

Serial Clock Register

The Serial Clock Register is used to specify the logic level of the serial clock output. Writing to bit[0] of this register places the written logic level on the serial clock pin. Reading bit[1] of the register returns the actual logic level of serial clock pin. All other bits in this register are unused.

SIM Interface

The BP provides a functional interface to the handset SIM card. Since the BP nominally operates at 2.8 V and produces 2.8 V logic levels, external voltage level translation circuitry is necessary to interface with a 5 V SIM card.

The SIM interface is a half duplex, serial synchronous data link, which is fully described in Sections 11.11 and 11.12 of the European Telecommunications Standard Institute (ETSI) GSM specifications.

The BP generates a number of SIM interface signals. Some are required by the ETSI definition of the SIM interface, after the external voltage translation circuitry, if required. Some implement auxiliary functions required for Skyworks's implementation of the SIM Interface. Table 35 provides a summary of these interface signals and their functions.

For the SIM signals that are multiplexed with GPIO signals, the appropriate I/O select bit must be set. See the GPIO Ports section for more information.

SIM Interrupts

Two different interrupts can be generated by the SIM interface circuitry:

- SIM timer interrupt
- SIM activity interrupt

SIM Timer Interrupt

The SIM timer interrupt is generated when a time out condition occurs. General-purpose timer A is used to detect that a time out has occurred. The timer can use the SIM system clock or the SIM Elementary Time Unit (ETU) clock as an input. See the Timers section for more information.

SIM Activity Interrupt

The SIM activity interrupt is generated when a recognized error has occurred on the SIM interface. Each of these interrupts can be individually enabled or disabled. They are described in the section on the SIM Interrupt Enable Register. If the SIM activity interrupt is generated, the ARM can read the SIM Status Register to determine the cause of the interrupt.

SIM Interface Registers

The following registers are described in this section. The address and default values for the SIM Interface Registers are specified in Table 6.

- SIM Control Register
- SIM Status Register
- SIM Interrupt Enable Register
- SIM Output Buffer
- SIM Input Buffer
- DMA Delay Count

SIM Control Register

The SIM Control Register controls the operation of the SIM interface. The register is 32-bits wide. The function of each of the bits in this register is described in Table 36.

Table 35. SIM Interface Signals

Signal	Direction	Function	Notes
SIM_DATA	I/O	SIM interface data signal	Required by ETSI specification
SIM_RESET	O	SIM interface reset signal (multiplexed with GPIO Port C[3])	Required by ETSI specification
SIM_CLOCK	O	SIM interface clock signal	Required by ETSI specification
SIM_ENABLE	O	Enable external circuitry to generate SIM supply (multiplexed with GPIO Port C[1])	Required by Skyworks implementation

Table 36. SIM Control Register Functions

Bit	Function
[31]	Reserved
[30]	DMA Delay enable: 0 = disabled 1 = enabled
[29]	Reset all the SIM Registers to their default values: 0 = reset 1 = normal operation
[28]	SIM_Clock divider ratio: 0 = divide by 1 1 = divide by 2
[27]	SIM_Receive Parity test enable: 0 = disabled 1 = enabled
[26]	SIM_Transmit Parity test enable: 0 = disabled 1 = enabled
[25]	SIM_Loop_Test enable: 0 = disabled 1 = enabled
[24:8]	SIM_etu_Factor. Divide ratio to generate SIM ETU from the SIM_Clock signal
[7]	SIM data transfer convention: 0 = direct convention 1 = inverse convention
[6]	Enable/disable SIM to ME retransmission: 0 = disabled 1 = enabled
[5]	Enable/disable ME to SIM retransmission: 0 = disabled 1 = enabled
[4]	SIM_CLOCK stop state 0 = stop at low level 1 = stop at high level
[3]	SIM_CLOCK on/off bit: 0 = off 1 = on
[2]	SIM_RESET active state: 0 = active low 1 = active high
[1]	SIM_DATA control: 0 = put SIM_DATA to low level 1 = normal
[0]	SIM enable bit: 0 = off 1 = on

SIM Status Register

The SIM Status Register provides information on the status of the SIM interface. If a SIM interrupt is received by the ARM, reading this register indicates the source of the interrupt. If the bit is set to a "1," the associated condition has occurred. The register is eight bits wide. The function of each of the bits in this register is described in

.

Table 37. SIM Status Register Functions

Bit	Function
[7:5]	Reserved
[4]	SIM output buffer empty
[3]	SIM input buffer full
[2]	SIM input buffer overrun error
[1]	SIM to ME receive failure
[0]	SIM to ME transmit failure

SIM Interrupt Enable Register

The SIM Interrupt Enable Register is used to mask the SIM interrupts. Each of the SIM interface conditions that can generate a SIM activity interrupt has an associated bit in this register which can be used to enable or disable the generation of the interrupt if the condition occurs. The register is eight bits wide. The function of each of the bits in this register is described in Table 38.

Table 38. SIM Interrupt Enable Register Functions

Bit	Function
[7:5]	Reserved
[4]	SIM output buffer empty interrupt enable/disable
[3]	SIM input buffer full interrupt enable/disable
[2]	SIM input buffer overrun error interrupt enable/disable
[1]	SIM to ME receive failure interrupt enable/disable
[0]	SIM to ME transmit failure interrupt enable/disable

SIM Output Buffer

The SIM Output Buffer is used to store the data to be transmitted over the SIM interface.

SIM Input Buffer

The SIM Input Buffer is used to store the data received over the SIM interface.

DMA Delay Count

The DMA Delay Count Register counts the number of 3.9 MHz clock delays for DMA requests.

DSP Core

The DSP core is a dedicated DSP processor core that implements all the physical layer, Layer 1, signal processing required by a GSM-based handset. The DSP core communicates with the controller core via the Dual Port RAM (DPRAM) memory. The DSP also interfaces to the IA.

Dual Port Memory

The DSP and ARM cores communicate via the 512 x 32-bit Dual Port Memory (DPRAM). On the ARM side, the DPRAM interfaces to the IPB. Instructions and information are passed between the two cores by writing to and reading from the device DPRAM.

DSP Memory

The DSP Countach Core supports FR, EFR and HR Speech Vocoder. Internal DSP memory consists of:

- PROM. 40 K x 32.
- DROM. 12 K x 32.
- Patch RAM. 512 x 32.
- Register Files. 1K x 32
- SPAD. 512 x 32
- Co-processor. 1K x 32.

DSP Interrupt Controller

The DSP core can handle two interrupts:

- Fast Interrupt Request (FIQ)
- Interrupt Request (IRQ)

The FIQ has a higher priority than the IRQ. The IRQ is masked when an FIQ sequence is entered. In the case of an FIQ interrupt, fewer registers are required to be saved to memory. Therefore, switching into the interrupt handler is slightly faster.

All possible interrupt sources, both internal and external, are routed to the Interrupt Controller, which generates either the FIQ or IRQ interrupt. The Interrupt Controller has the following features:

- Supports 32 fixed interrupt sources
- Programmable level or edge sensitivity for each interrupt source
- Programmable input polarity for each interrupt source
- Provides a mapping to the FIQ or IRQ outputs
- Programmable interrupt enables mask
- Programmable test interrupts enabling embedded verification and test
- Six-bit Priority Encoded Interrupt Vector generated as output

Interrupt Controller Registers

The following registers are described in this section. The address and default values for the Interrupt Controller Registers are specified in Table 7.

- Interrupt Pending Register
- Interrupt Select Register
- Interrupt Enable Register
- Interrupt Type Register
- Interrupt Polarity Register
- Interrupt Condition Register
- FIQ Interrupt Register
- IRQ Interrupt Register

Interrupt Pending Register

All interrupt sources are latched into the Interrupt Pending Register. When an interrupt is latched into this register, the bit remains set to “1” until the interrupt source has disappeared and the DSP clears the bit. The source of each of the bits in the register is specified in Table 10.

Interrupt Select Register

Every enabled interrupt source can generate either an FIQ or IRQ interrupt to the DSP core. The Interrupt Select Register contains a bit for each possible interrupt source.

- If the associated bit is set to “1”, an FIQ interrupt is generated when an interrupt occurs and the interrupt is enabled.
- Conversely, if the bit is set to “0”, an IRQ interrupt is generated when an interrupt occurs and the interrupt is enabled.

The Interrupt Select Register bits have the same mapping to the interrupt sources as the Interrupt Pending Register, Table 39.

Interrupt Enable Register

The Interrupt Enable Register contains a corresponding bit for each possible interrupt source. If the bit is set to “1,” and an interrupt occurs, an interrupt is sent to the DSP. Either an FIQ or IRQ interrupt is generated, depending on the status of the associated interrupt bit in the Interrupt Select Register. If the bit is set to “0,” the interrupt is disabled.

The Interrupt Enable Register bits have the same mapping to the interrupt sources as the Interrupt Pending Register, Table 39.

Interrupt Type Register

This register defines the edge or level triggered type interrupts. If the bit is set to “1”, the corresponding interrupt is edge-sensitive; otherwise a level is detected. An edge-sensitive interrupt can be a “1” or “0” level without causing an interrupt request. Only the actual edge causes an interrupt request.

Table 39. Interrupt Pending Register Sources

Cntlr	Intrpt #	Interrupt Name	Cntlr	Intrpt #	Interrupt Name
Intc	31	Unused	Intc	15	Cidex
Intc	30	Unused	Intc	14	Viterbi
Intc	29	Unused	Intc	13	Host_IRQ(3)
Intc	28	Unused	Intc	12	Host_IRQ(2)
Intc	27	Unused	Intc	11	Host_IRQ(1)
Intc	26	Unused	Intc	10	Host_IRQ(0)
Intc	25	Unused	Intc	9	Frame_Tick
Intc	24	Unused	Intc	8	Timer
Intc	23	NOT DAI_RST	Intc	7	USART # 4. RX
Intc	22	DAI_RST	Intc	6	USART # 4. TX
Intc	21	Software IRQ # 3	Intc	5	USART # 2. RX
Intc	20	Software IRQ # 2	Intc	4	USART # 2. TX
Intc	19	Software IRQ # 1	Intc	3	USART # 3. RX
Intc	18	Wake-up	Intc	2	USART # 3. TX
Intc	17	AMBA_DMA_IRQ(1)	Intc	1	RX Port . DMA Bank 1
Intc	16	AMBA_DMA_IRQ(0)	Intc	0	Rx Port . DMA Bank 0

Interrupt Polarity Register

Writing to the appropriate Interrupt Polarity Register bit selects the polarity of all external interrupts.

- If the bit is set to "1," an interrupt is generated either on the rising edge for edge mode or on the high level for level mode.
- If the bit is set to "0," an interrupt is generated either on the falling edge for edge mode or on the low level for level mode.

Caution: *Care must be taken since the act of altering the bit could result in the generation of an interrupt edge. This potential hazard can be avoided by using software to disable the interrupt source when the polarity bit is changed.*

The External Interrupt Polarity Register bits have the same mapping to the interrupt sources as the Interrupt Pending Register, see Table 10. For internally generated interrupts, the associated bits in this register are unused.

Interrupt Condition Register

This register defines the corresponding interrupts to be synchronous or asynchronous detected.

- If the bit is set to "1", the interrupt is synchronous-detected
- Otherwise, an asynchronous is assumed.

FIQ Interrupt Register

The FIQ Interrupt Register contains bits for all the possible interrupt sources. The FIQ Interrupt Register bits have the same mapping to the interrupt sources as the Interrupt Pending Register, Table 39.

- If a bit for a particular interrupt is set to "1", the following conditions apply to that interrupt:
  - The interrupt has occurred
  - The interrupt is enabled
  - The interrupt is set to generate an FIQ interrupt
- If the bit is set to "0", at least one of the conditions listed above is not met.

IRQ Interrupt Register

The IRQ Interrupt Register contains bits for all the possible interrupt sources. The IRQ Interrupt Register bits have the same mapping to the interrupt sources as the Interrupt Pending Register, Table 39.

- If a bit for a particular interrupt is set to "1", the following conditions apply to that interrupt:
  - The interrupt has occurred
  - The interrupt is enabled
  - The interrupt is set to generate an IRQ interrupt
- If the bit is set to "0", at least one of the conditions listed above is not met.

Trap and Patch

The Trap and Patch block performs the following peripheral functions:

- Provides 16 Trap Registers, which Trap, or intercept, the DSP Program Counter.
- Provides a means to Trap on certain “other, non-PC” type events, for example, DSP IRQs, Data Addresses. Those events include DSP Opcodes, DRAM Addresses (S0, S1, Dst), and Interrupts.

Trap and Patch

The concept is to selectively monitor the DSP Program Count, and check for particular addresses. Once found, these addresses simply cause a jump to Program RAM, which is 512 words in ARCTIX. That RAM has a vector table which again jumps to the Program RAM (PRAM), and executes some code. When this other code is complete, the return is to the address following the final address executed.

Trap and Freeze

The Trap & Freeze goal is to stop all DSP activity, if desired, at a given point. When triggered, the Trap & Freeze feature stops all hardware activity on the DSPSIDE, and allows the ARM to come and dump the contents for debug.

Trap and Patch Registers

The following registers are described in this section. The address and default values for the Trap & Patch Registers are specified in Table 7.

- Trap Enable A Register
- Trap Enable B Register
- Trap Address Registers
- Trap Data Bus Address Register
- Last Address Registers
- Trap Opcode Registers
- Trap & Patch Status Register

Trap Enable A Register

This Trap Enable Register specifies which 16 trap address registers will be compared to the PC. Each bit is corresponding to the associated trap address register if enable.

Trap Enable B Register

This Trap Enable Register specifies the operation mode of Trap & Patch block. The function of each of the bits in the register is shown in Table 40.

Table 40. Trap Enable B Register Functions

Bit	Function
[15:8]	Interrupt Selection for Trap Bit[8]. Any of USART Rx_IRQ Bit[9]. Cidex IRQ Bit[10]. Viterbi IRQ Bit[11]. Frame Tick from IA Bit[12]. DSP Timer IRQ Bit[13]. AMBA DMA IRQ Bit[14]. DSP SW IRQ Bit[15]. ARM2DSP IRQ
[7]	Reserved
[6]	Trap & Freeze Mode Enable 0 = disabled 1 = enabled
[5:4]	Reserved
[3]	Trap for Opcode Enable 0 = disabled 1 = enabled
[2]	Trap for Data bus transaction Enable 0 = disabled 1 = enabled
[1]	Trap on Patch event Enable 0 = disabled 1 = enabled
[0]	Trap or Interrupt Enable 0 = disabled 1 = enabled

Trap Address Registers

There are sixteen 16-bit Address Registers. Each of them contains the address to be compared with PC if enabled in Trap Enable B Register.

Trap Data Bus Address Register

This register contains the data bus address at which the trap event should occur.

Last Address Registers

These two registers contain the last value of the PC when a trap event occurred.

Trap Opcode Registers

These two registers contain the DSP Opcode at which the trap event should occur.

Trap & Patch Status Register

The Trap & Patch Status Register reports the operation of the Trap & Patch block. The function of each of the bits in the register is shown in Table 41.

Table 41. Trap & Patch Status Register Functions

Bit	Function
[15:9]	Reserved
[8:4]	Last Patch
[3]	Opcode Driven Trap: 0 = no 1 = occur
[2]	Data Bus Driven Trap: 0 = no 1 = occur
[1]	PC Driven Trap: 0 = no 1 = occur
[0]	Interrupt Driven Trap: 0 = no 1 = occur

Cidex

Cidex is a coprocessor engine for the DSP Core to perform the Cidex 1 or Cidex 2 algorithm. These algorithms perform ciphering data on each frame data in GSM circuit switch mode. The Cidex block has two main modes:

- Cidex Algorithm 1
- Cidex Algorithm 2

Each algorithm generates two 114-bit binary sequences, BLOCK1 and BLOCK2, from a 64-bit cipher key, Kc, a 22-bit time variable, COUNT, and the algorithm mode bit, MODE.

This engine has the following operation features:

- Number of Input bits. 1 + 64 + 22 = 87
- Number of Output bits. 114 + 114 = 228
- Number of clock cycles for a run. 86 + 100 + 228 + 26 = 440

A typical Cidex data flow diagram is shown in Figure 18.

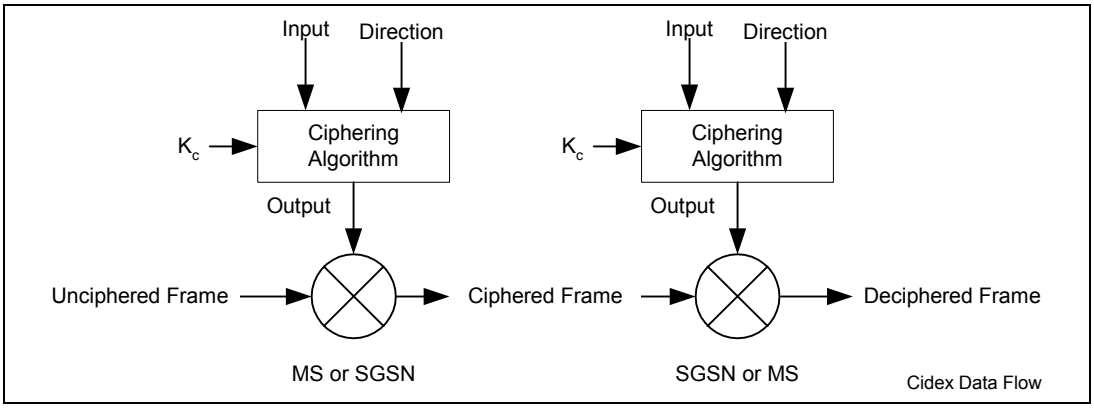


Figure 18. Cidex Data Flow Diagram

Cidex Registers

The following registers are described in this section. The address and default values for the Cidex Registers are specified in Table 7.

- Cidex Control Register
- Cidex Status Register
- Key Input Word Registers
- TDMA Word Registers
- Block 1 Registers
- Block 2 Registers

Cidex Control Register

The Cidex Control Register configures the operation of the Cidex block. The function of each of the bits in the register is shown in Table 42.

Table 42. Cidex Control Register Functions

Bit	Function
[15:5]	Reserved
[4]	Cidex algorithm select: 0 = Cidex 1 1 = Cidex 2
[3]	Go: 0 = wait 1 = start encipher/decipher
[2]	Interrupt Enable: 0 = disabled 1 = enabled
[1]	Cidex block Enable: 0 = disabled 1 = enabled
[0]	Reset: 0 = reset Cidex block 1 = normal

Cidex Status Register

The Cidex Status Register reports the operation of the Cidex block. The function of each of the bits in the register is shown in Table 43.

Table 43. Cidex Status Register Functions

Bit	Function
[15:2]	Reserved
[1]	Clock request activity: 0 = idle 1 = active
[0]	Cidex activity: 0 = idle 1 = busy

Key Input Word Registers

There are four 16-bit Key Input Registers, which make up the 64-bit key for the ciphering engine.

TDMA Word Registers

There are two 16-bit Input Registers, which make up the 22-bit time variable data for the ciphering engine.

Block 1 Registers

There are eight 16-bit output registers, which make up the 114-bit binary sequence output data.

Block 2 Registers

There are eight 16-bit output registers, which make up the 114-bit binary sequence output data.

Viterbi

The Viterbi block provides a coprocessor engine to perform the Viterbi Convolutional and Maximum Likelihood Sequence Estimation (MLSE) Algorithms. The algorithm works with the following factors:

- Up to 32 states with a constraint length of 6
- A code rate of 1/2 to 1/6
- A maximum traceback length of 30
- A maximum symbol count of 1K

The Viterbi block operates in the following modes. The mode and number of input/output data words determine length of a run.

- Convolutional, that is, Full-rate, half-rate
- MLSE

The block has the following characteristics:

- Inputs
  - 12-bits per word, Symbol/FS data input via the Coproc Dual-port RAM
  - In MLSE mode, the FS data must be before the input symbols, while in Convolutional mode, the input symbols start from Start_addr, as there is no FS-values.
  - Starting address of input data in RAM, Start_addr
  - Algorithm Mode (MLSE or Conv) bit, Mode
  - Table Length
  - Code Rate
  - Number of States
  - Initial and Ending States
  - Number of symbols
  - Code Polynomials 1 through 6
  - Requires twelve DPB writes to program and begin a run.
- Outputs
  - 12-bits per word, output data via the Coproc_Mux to the Dual-port RAM

Note. *The output data is written back to the Dual-port RAM starting at Start_Addr + Log_2(Num States)*

- External Interfaces
  - The Viterbix interfaces to the DSP Peripheral Bus (DPB) and the Coproc Dual-port RAM via the CoProc_Mux.
  - Viterbix provides an optional DSP interrupt indicating that the run is complete (or activity is complete when in test mode).
  - Viterbix also provides a clock request signal that requests a DSP clock only when needed.

Viterbi Registers

The following registers are described in this section. The address and default values for the Viterbi Registers are specified in Table 7.

- Viterbi Control Register
- Viterbi Status Register
- Viterbi Configuration Register
- Starting Address Data RAM Register
- Initial/Ending State Register
- Symbols Register
- Code Polynomial Registers
- RAM Test Address Register
- RAM Test Data Input Registers
- RAM Test Data Output Registers

Viterbi Control Register

The Viterbi Control Register configures the operation of the Viterbi block. The function of each of the bits in the register is shown in Table 44.

Table 44. Viterbi Control Register Functions

Bit	Function
[15:11]	Reserved
[10:9]	Read/Write Operation: 00 = Wait 01 = Read from programmed address 10 = Write data to programmed address 11 = no action
[8:7]	RAM Selection: 00 = select TB1_RAM (hard data) 01 = select TB2_RAM (soft data) 10 = select Metric_RAM 11 = select Metric_RAM
[6]	RAM Test Enable: 0 = disabled 1 = enabled
[5:3]	Operation condition: 000 = wait 001 = Go, continous mode 010 = Step, DSP control 011 = Init_Step 100 = Flush
[2]	Interrupt Enable: 0 = disabled 1 = enabled
[1]	Viterbi block Enable: 0 = disabled 1 = enabled
[0]	Reset: 0 = reset Viterbi block 1 = normal

Viterbi Status Register

The Viterbi Status Register reports the operation of the Viterbi block. The function of each of the bits in the register is shown in Table 45.

Table 45. Viterbi Status Register Functions

Bit	Function
[15:7]	Reserved
[6:5]	RAM Test activity: 00 = idle 01 = Test Read in progress 10 = Test Write in progress 11 = not valid
[4]	Clock request activity: 0 = idle 1 = active
[3]	Flush activity: 0 = idle 1 = busy
[2]	Init_Step activity: 0 = idle 1 = busy
[1]	Step activity: 0 = idle 1 = busy
[0]	Go activity: 0 = idle 1 = busy

Viterbi Configuration Register

The Viterbi Status Register specifies the configuration setup of the Viterbi block. The function of each of the bits in the register is shown in Table 46.

Table 46. Viterbi Configuration Register Functions

Bit	Function
[15:10]	Reserved
[10:9]	Number of states: 00 = 16 states 01 = 32 states 10 = 64 states 11 = 64 states
[8:6]	Code rate
[5:1]	Table length
[0]	Mode: 0 = convolutional decoder 1 = MLSE equalizer

Starting Address Data RAM Register

This register specifies the starting address of Data in RAM. Only bit[9:0] are used, bit[15:11] are unused.

Initial/Ending State Register

This register specifies the starting state and ending state. Bit [5:0] indicate ending state, bit [11:6] indicate starting state and bit[15:11] are unused.

Symbols Register

This register specifies the number of symbols. Only bit[9:0] are used, bit[15:11] are unused.

Code Polynomial Registers

There are six 16-bit Registers. Only bit[5:0] are used, bit[15:6] are not used.

RAM Test Address Register

This register specifies the address of RAM test. Only bit[6:0] are used, bit[15:7] are unused.

RAM Test Data Input Registers

There are two 16-bit Registers that make up the 24-bit input data for RAM test. If Met RAM is selected, 24-bit data is used. Other RAMs use 12-bit Least Significant Bit (LSB).

RAM Test Data Output Registers

There are two 16-bit Registers that make up the 24-bit output data for RAM test. If Met RAM is selected, 24-bit data is used. Other RAMs only use 12-bit LSB.

Timer

A single 26-bit general purpose counters/timer is used to generate time-related interrupts to the DSP Core. Its input clock can be either the system clock, 3.9 MHz, or the DSP clock. The Timer consists of a Latch Register and a Counter Register. The Counter Register is loaded from the Latch Register and the timer counts down the contents of the Counter Register. When the Counter Register contents reach 0, the interrupt is generated.

The Timer has a dedicated Timer Control Register. Register contents determine timer configuration. The address and default values for the Viterbi Registers are specified in Table 7. Register bit functions are provided in Table 47.

Table 47. Timer Mode Register Functions

Timer Control Register Data																
offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0C	Reserved									ESN	TLS	INE	TMS	CKS	MOD	

Read value after reset = 0x0008

Bit[5] TLS (Timer Load Select)
- 0 = Data written to TPRM/TPRL or TIMM/TIML is not used by the timer until both MS and LS halves are loaded. This ensures the timer gets loaded with the entire 26-bit at the same time. Therefore, it is not necessary to stop the timer when changing TPR or TIM values.
- 1 = Data written to TPRM/TPRL or TIMM/TIML will be loaded into the corresponding bits in the timer immediately after synchronization. This mode can be used if only half of the TPR or TIM needs to be changed.

Bit[4] INE (Interrupt Enable)
- 0 = disable interrupt output generation
- 1 = enable interrupt output generation

Bit[3] TMS (Timer Stop)
- 0 = normal condition, timer counts
- 1 = stops timer clock.

Bit[2] CKS (Clock Select) *
- 0 = timer uses input clock CLKIN1
- 1 = timer uses input clock CLKIN2

Bit[1:0] MOD (Mode) *

2 bits select the timer operating mode as follows:
- 00 = Interval Timer
- 01 = Pulse Generator
- 10 = Event Counter
- 11 = Pulse Width Measurement

Control Interface

The control interface is a four-wire serial interface that provides an interface between the BP and the Skyworks IA device. The interface is a high speed, synchronous, full duplex, serial communications link. The interface is connected to the DSP of the BP.

The control interface consists of the following signals:

- CNTRLCLK. 3.9 MHz clock signal output from the BP. This is the clock signal for the interface.
- CNTRLRT. Indicates the start and end of a communications session. This signal is output from the BP.
- CNTRLDAT. Serial output data from the BP.
- RSPNSDAT. Serial data input to the BP.

The BP is the bus master for the interface. It initiates all communications over the interface. Using this port, the BP can perform the following functions:

- Send control information to configure the IA operation.
- Send bursts of transit data to the IA device for modulation.
- Read contents of the IA Registers.

Figure 19 shows the signal sequence for write and read operations over the control interface.

Codec Interface

The Codec interface is a four-wire serial interface that provides an interface between the BP and the IA device. The interface is a high speed, synchronous, full duplex, serial communications link. The interface is connected to the DSP of the BP.

The Codec interface consists of the following signals:

- CDCCLK. 4 MHz input clock
- CDCRATE. 8 kHz input framing signal
- ENCDRDAT. Serial data input to the BP. The bit rate is the same as the CDCCLK rate, 4 Mbps. The word rate is the same as the CDCRATE signal, 8 kwps. Words are 16 bits wide.
- DCDRDAT. Serial data output from the BP. The bit rate is the same as the CDCCLK rate, 4 Mbps. The word rate is the same as the CDCRATE signal, 8 kwps. Words are 16 bits wide.

When a voice call is in progress, the following occurs:

- In the receive path, the BP sends out digitized audio samples using the Codec interface. The IA converts these digitized samples to an analog signal, They are then used to drive the handset speaker.
- In the transmit path, the IA converts the analog output from the handset microphone into digital samples that are sent to the BP using the Codec interface. The DSP processes the samples for transmission by the handset RF subsystem.

Figure 20 shows BP Codec interface timing diagram.

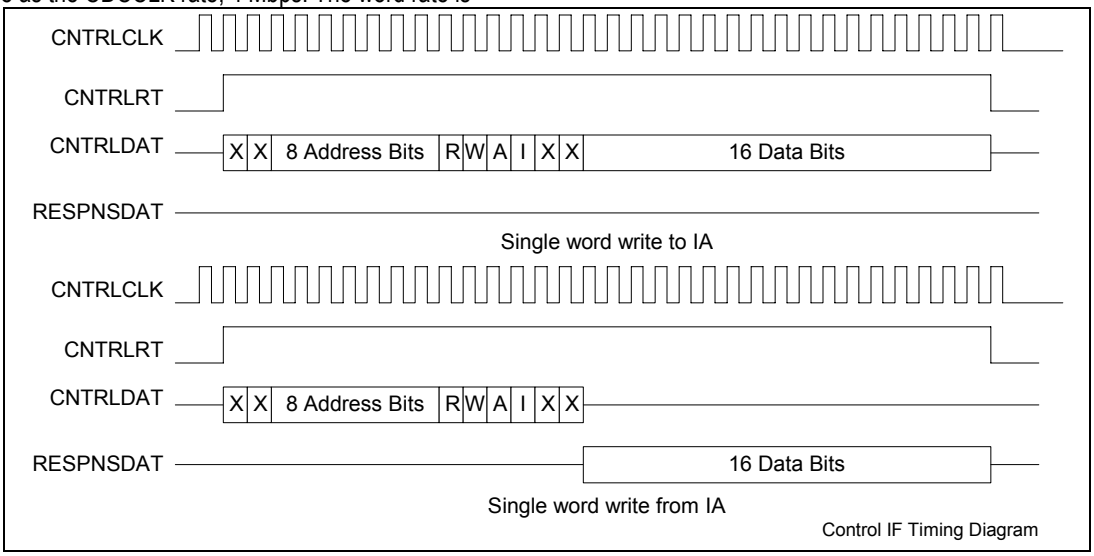


Figure 19. Control Interface Timing Diagram

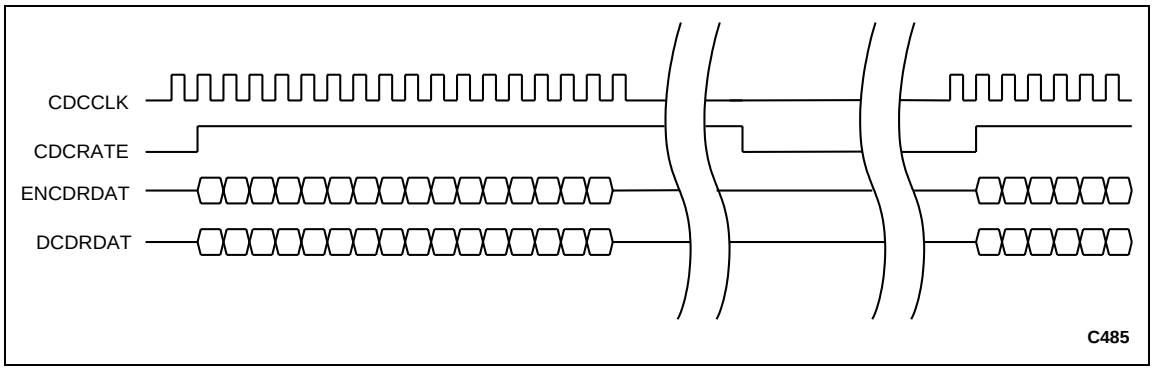


Figure 20. Codec Interface Timing Diagram

Receive Interface

The receive interface is a three-wire serial interface that provides an interface between the BP and the IA device. The interface is a high speed, synchronous, simplex, serial communications link. Data transfer direction is from the IA to the BP. The interface is connected to the BP DSP. Samples received over the interface are stored in dedicated RAM for processing by the DSP.

The receive interface consists of the following three signals:

- RX_CLOCK. 19.5 MHz clock input.
- RX_RATE. 1.0833 MHz word rate input signal.
- RX_DATA. serial data input to the BP. The data rate is 19.5 Mbps.

Digitized In-Phase and Quadrature (I/Q) samples, recovered from the received RF signal, are sent from the RF subsystem to the BP over this interface. The samples are stored in RAM for further processing by the DSP.

Figure 21 shows the timing diagram for the BP receive interface.

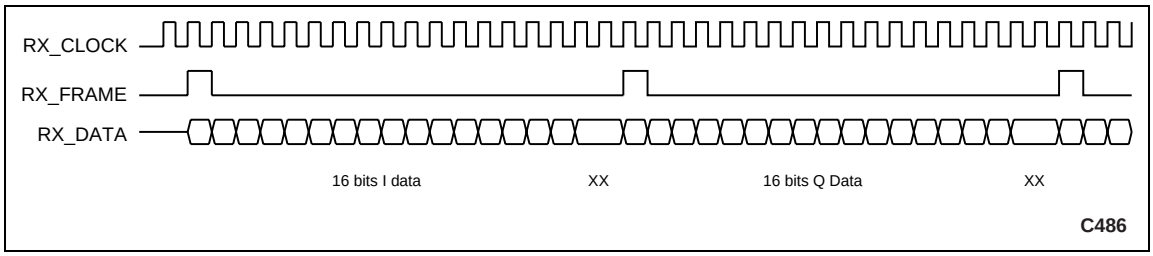


Figure 21. BP Receive Interface Timing Diagram

Digital Audio Interface Port

The ETSI specifications for Type Approval testing of a GSM handset, GSM 11.10, require a Digital Audio Interface (DAI) port to verify the handset audio paths and transducers, for example, microphone, speaker. During Type Approval testing, the DAI port interfaces to the System Simulator test equipment.

The DAI Interface is a full duplex serial interface consisting of the following signals:

- DAI_CLOCK. DAI interface clock output signal
- DAI_RX. DAI interface receive data input signal
- DAI_TX. DAI interface transmit data output signal
- DAI_RESET. DAI interface reset input signal

The DAI signals are multiplexed with GPIO pins on the BP device. To enable the DAI functionality of the pins, a DAI_Enable bit is set to "1" in one of the device registers.

Table 48 specifies which DAI signals are multiplexed with GPIO pins.

Table 48. DAI Signal Multiplexing With GPIO Pins

GPIO Pin	DAI Signal
Port B[3]	DAI_CLOCK
Port C[6]	DAI_RX
Port C[5]	DAI_TX
Port B[6]	DAI_RESET

DSP Debug Interface

This serial debug interface is the UART port to output Layer 1 DSP trace information for debugging purposes. The port (DSP_Rx and DSP_Tx) operates in 8-bit asynchronous mode, which allows it to connect to a data terminal or PC. DSP Core can also use two GPIO bits to exchange command/status information with external data terminal. These DSP trace signals are multiplexed with GPIO pins on the BP device. To enable the debug functionality of the pin, a FldTest bit is set to "1" in the Test Control Register # 3.

Table 49 specifies which DSP trace signals are multiplexed with GPIO pins.

Table 49. DSP Trace Multiplexing With GPIO Pins

GPIO Pin	Trace Signal
Port A[2]	DSP GPIO bit 1
Port C[7]	DSP GPIO bit 0
Port D[0]	DSP Trace Rx
Port D[2]	DSP Trace Tx

Electrical and Mechanical Specifications

The absolute maximum ratings of the BP are provided in Table 50. The recommended operating conditions are specified in Table 51. The electrical characteristics are specified in Table 52.

Note: *The BP device has digital outputs with different output drive current capabilities.*

Table 53 specifies the output drive level of each output pin. Figure 22 shows a block diagram for a typical application of the device. Figure 23 provides the package dimensions for the 160-pin FPBGA. Figure 24 provides the shipping tray dimensions, and Figure 25 provides tape and reel dimensions. Figure 26 provides marking layout details..

Table 50. Absolute Maximum Ratings

Parameter	Symbol	Limits	Units
Supply voltage	Vdd	3.6	V
Input voltage	V_{IN}	3.6	V
Static discharge voltage (25 °C)	V_{ESD}	Human Body Model. ±4500 Charged Device Model. ±500	V
Latch-up current (25 °C)	I_{TRIG}	±150	mA
Storage temperature range	T_{STG}	–55 to +150	°C
***Note**. Voltages referenced to ground (Vss).*			

Table 51. BP Device Recommended Operating Conditions

Parameter	Symbol	Limits	Units
Supply voltage	Vdd	+2.7 to +3.6	V
Operating ambient temperature range	TA	–40 to +85	°C
RTC supply voltage	V_{rtc}	+2.3 to +2.8	V

Table 52. BP Device Digital Electrical Characteristics

Parameter	Symbol	Test Conditions	Min	Max	Units
Input High voltage (logic 1)	V_{IH}	I = 10uA (max)	1.96	Vcc+0.3	V
Input low voltage (logic 0)	V_{IL}	I = - 10uA (max)	-0.3	0.84	V
Output high voltage (logic 1)	V_{OH}	I = 2000uA (max)	2.38	Vcc+0.3	V
Output low voltage (logic 0)	V_{OL}	I = - 2000uA (max)	-0.3	0.42	V
Rise time	Tr	Cload = 15pF		5	ns
Fall time	Tf	Cload = 15pF		5	ns
***Note:** All voltages referenced to ground (Vss). Currents are positive when flowing into the device.*					

Table 53. Digital Output Drive Levels

Pin	Drive Level (mA)	Pin	Drive Level (mA)
A[23:0]	6	CNTRLDAT	4.8
BS[1:0]	6	CNTRLRT	4.8
D[15:0]	6	CS[7:0]	4.8
READ	6	DCDRDAT	4.8
WRITE	6	GPIO_A[7:0]	Programmable, 0.8 – 16.8
GPIO_C[3]	Programmable, 0.8 – 16.8	GPIO_B[7:0]	Programmable, 0.8 – 16.8
SIM_CLK	4.8	GPIO_C[2:0]	Programmable, 0.8 – 16.8
SRLCLK	4.8	GPIO_C[7:4]	Programmable, 0.8 – 16.8
SRLDATA	4.8	GPIO_D[2:0]	Programmable, 0.8 – 16.8
TDO	4.8		
ALARM	4.8		
CLK_REQ	4.8	KPDSTB[7:0]	4.8
CNTRLCLK	4.8	SDS_TX	4.8
		SIM_DATA	16

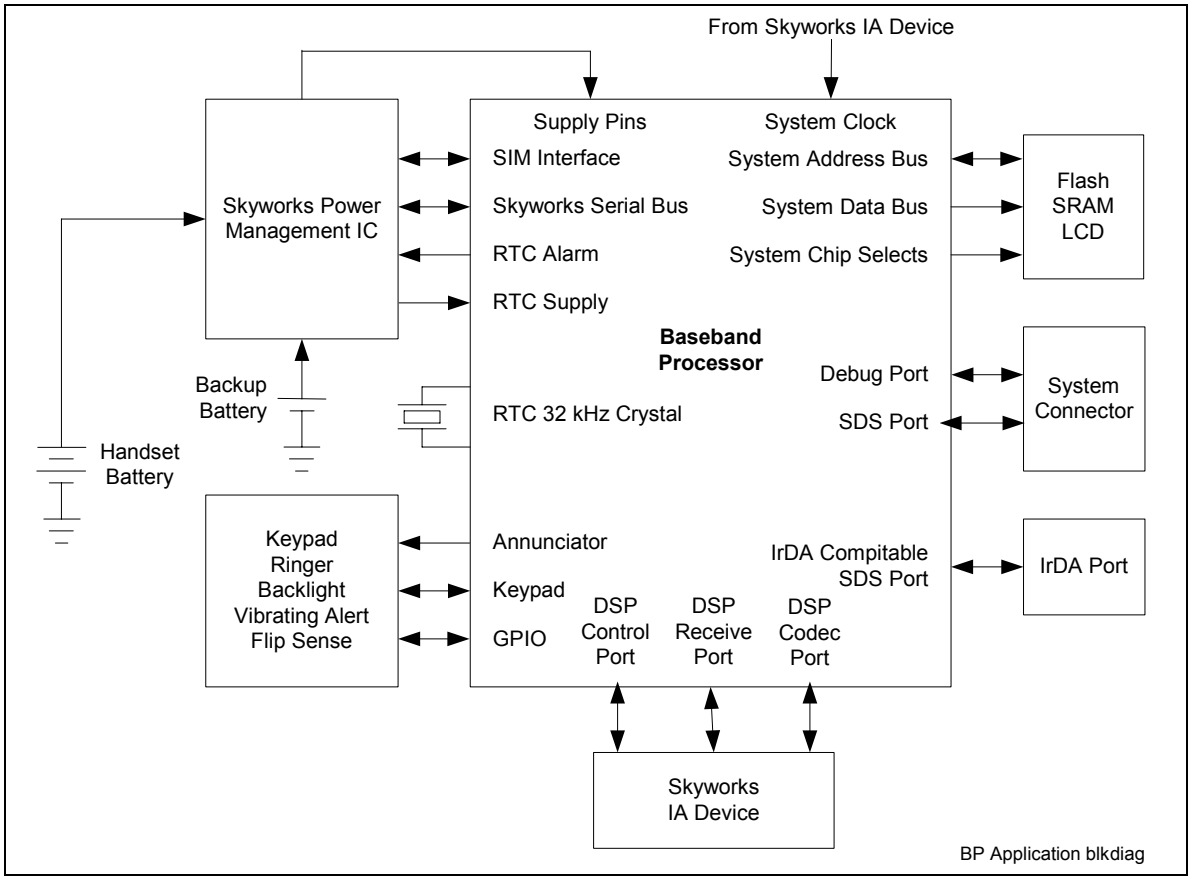


Figure 22. Typical BP Application Block Diagram

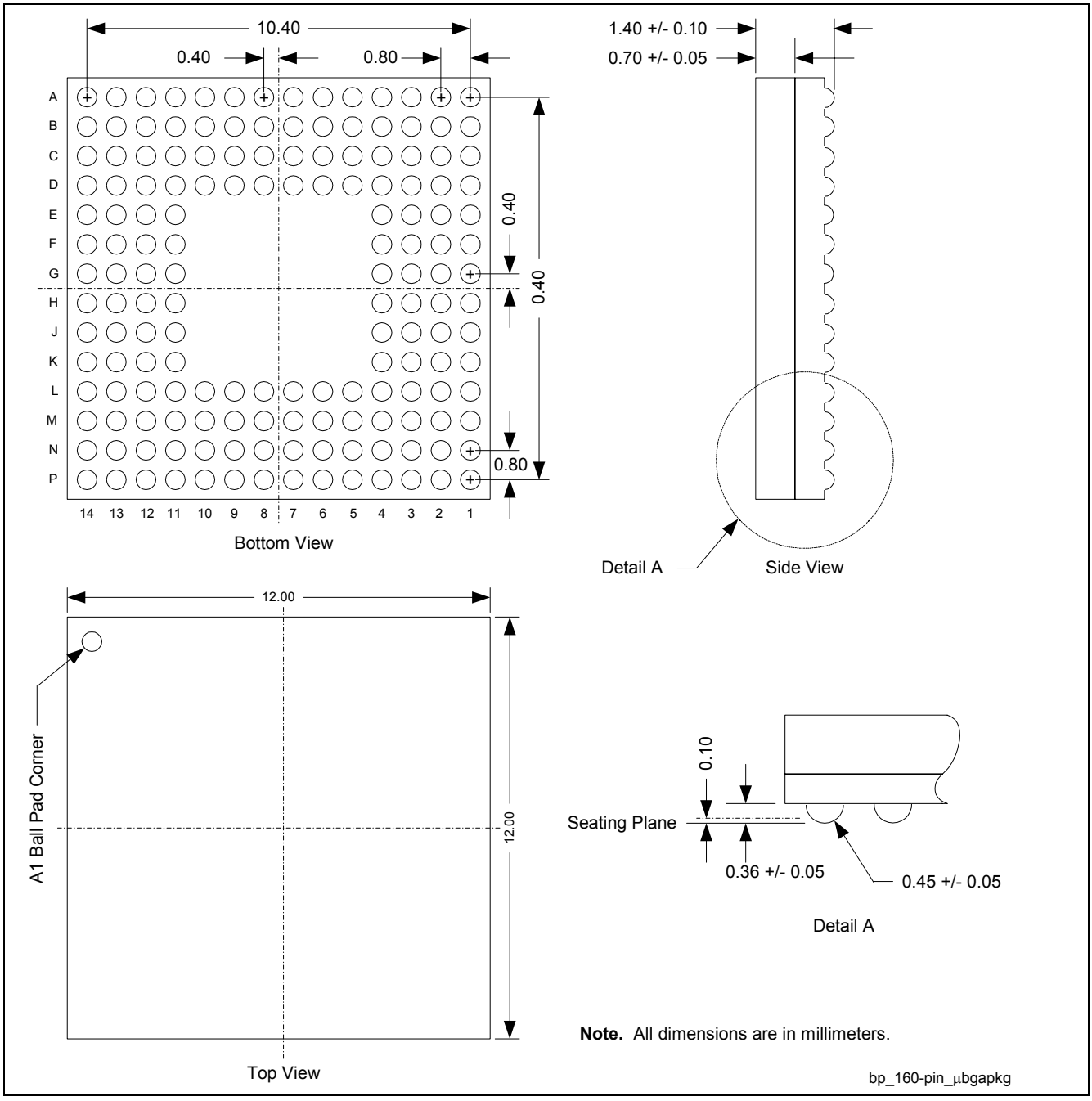


Figure 23. BP 160-Pin FPBGA Package Dimensions

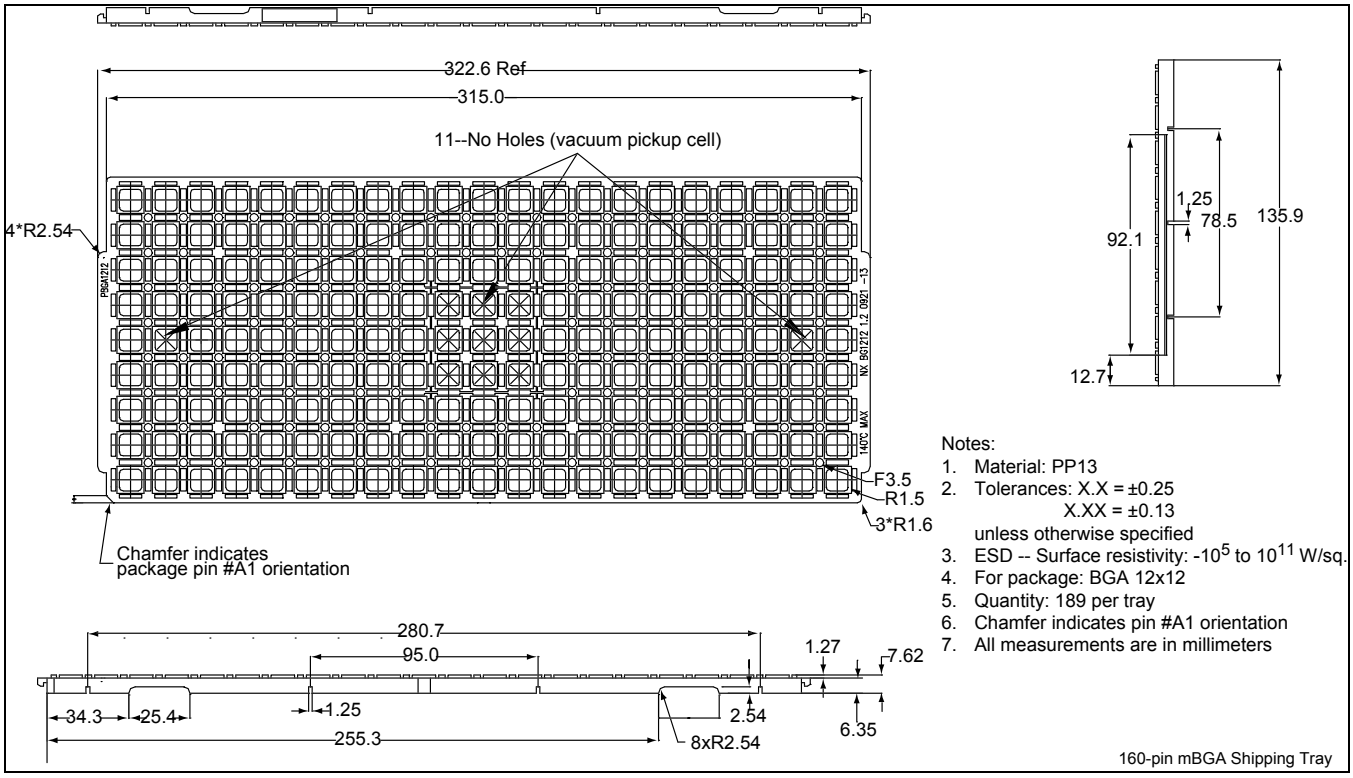


Figure 24. BP 160-Pin FPBGA Shipping Tray Dimensions

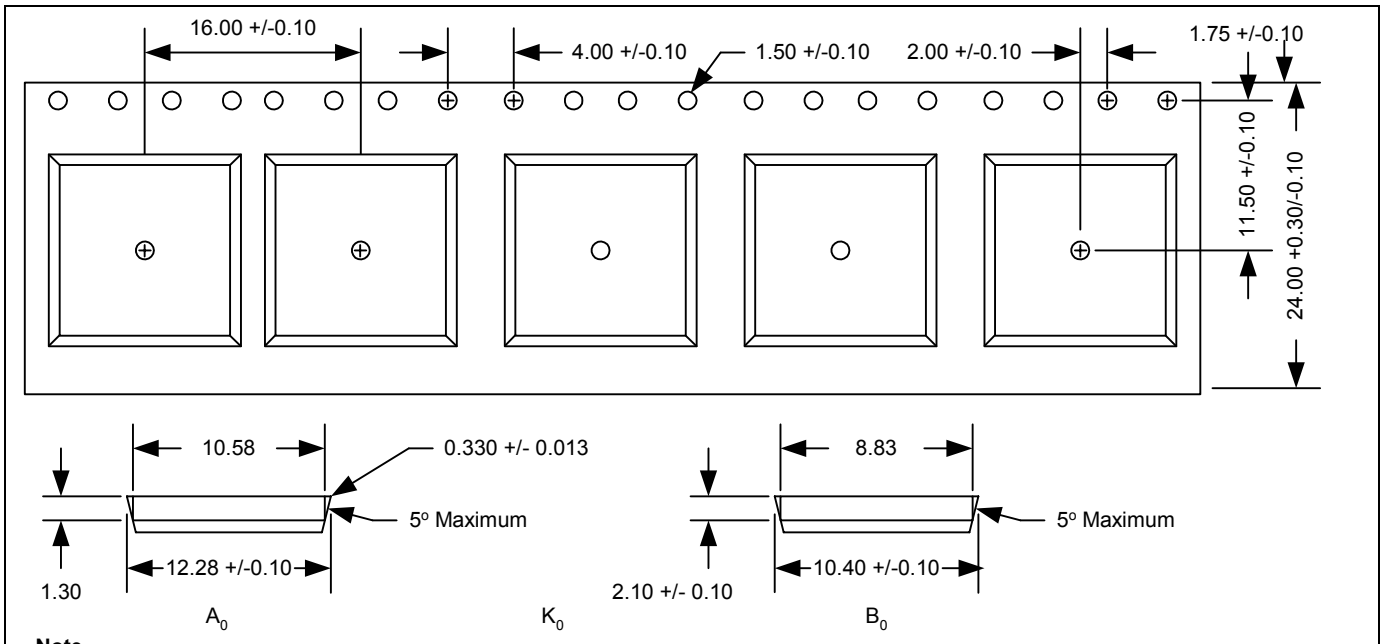
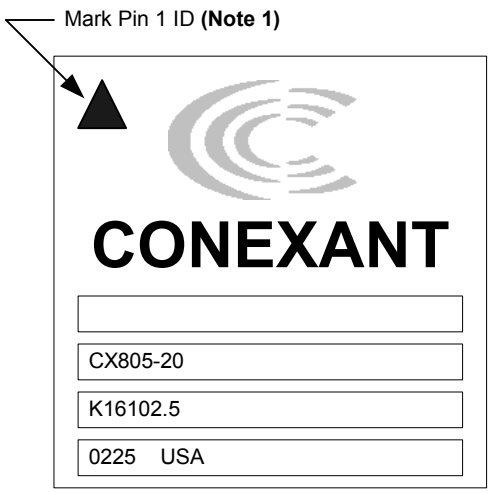


Figure 25. BP 160-Pin FPBGA Tape and Reel Dimensions



Note 1. The Pin 1 ID is a triangle or circle.

Note 2. **Brand line 1.** The Product Description is optional. It can be abbreviated to package requirements, but is not a derivative of the Part Number. **(Note 3)**

Brand line 2. The Part Number format is CXPPPPP-DD. The CX prefix is 2-alphabetic character company identifier. P is the 3- or 5-digit Part Number; D is a 2-digit dash number, for example, -20, -31. The CX prefix may not appear on small devices. The part number may be followed by a "P" to indicate a prototype device. **(Note 3)**

Brand line 3. Lot number and Lot Split Identifier. The Lot Number format is six alphanumeric characters followed by a 1- or 2-digit Lot Split Identifier. These are separated by a decimal point. The format is A12345.2 or A12345.21. **(Note 3)**

Brand line 4. Date Code and Country of Origin. The Date Code should be the same for the entire Lot Number and Lot Split Identifier. The first two digits of the Date Code are the current accounting calendar year. The last two digits are the current accounting calendar week. The format is YYWW, for example, 0225.
The Country of Origin is the full name of the country where assembly is completed, for example, Mexico. The country of origin may be abbreviated, for example, USA, CN, if backside marking is not possible because of size restrictions. **(Note 3)**

A vendor-specified logo may appear below Brand line 4, for example, ARM.

Note 3. As long as the device form, fit, and function remain the same, the data in Brand lines 1-4 may change. For example, the Lot Number and Lot Split Identifier may change; the Date Code and Country of Origin may change as Skyworks may select a second assembly source.

Marking Layout

Figure 26. CX805-20 Marking Layout

Ordering Information

Table 54. Ordering Information

Model Name	Part Number	DSP Code Version	Comments
Baseband Processor	CX805-20	TBD	TBD

General Information:

Skyworks Solutions, Inc.
4311 Jamboree Rd.
Newport Beach, CA 92660-3007

www.skyworksinc.com

SKYWORKS™
BREAKTHROUGH SIMPLICITY