Features

- Dual Core System Integrating an ARM7TDMI™ ARM® Thumb® Processor Core and a mAgic DSP for Audio, Communication and Beam-forming Applications
- High Performance DSP Operating at 100 MHz
  - 1 GFLOPS - 1.5 Gops
  - 10 Arithmetic Operations per Cycle (4 Multiply, 2 Add/subtract, 1 Add, 1 Subtract Floating and Fixed Point) Allowing Single Cycle FFT Butterfly
  - Native Support for Complex Arithmetic and Vectorial SIMD Operations: One Complex Multiply with Dual Add/sub per Clock Cycle or Two Real Multiply and Two Add/sub or Simple Scalar Operations
  - 32-bit Integer and IEEE 40-bit Extended Precision Floating Point Numeric Format
  - Large Multi-port Data Register File: 512 Registers Organized in Two 4-input 4-output 256-register Banks
  - Orthogonal VLIW Architecture, Code Compression for Code Size Reduction
  - Flexible Addressing Capability: 2 Independent Address Generation Units Operating on a 16 Registers Address Register File Supporting Programmable Stride, Circular Pointers and Bit Reversal
  - 1.7 Mbits of On-chip SRAM:
    17 K x 40-bit Data Memory Locations
    8 K x 128-bit Program Memory Location, Equivalent to 24K Instructions
  - DMA Access to the External Program and Data Memory
  - Two Main Operating Modes: Run and System Mode
  - Efficient Optimizing Assembler: Allows Easy Exploitation of the Available Hardware Resources Parallelism
- Utilizes the ARM7TDMI™ Processor Core with 32 K Byte of Integrated SRAM, Operating at 50 MHz
  - Fully-programmable External Bus Interface (EBI)
    Maximum External Address Space of 4 M Bytes
    Up to 4 Chip Selects
    Software-programmable 8/16-bit External Data Bus
  - 8-channel Peripheral Data Controller (PDC)
  - 8-level Priority, Individually Maskable Vectored Interrupt Controller
    4 External, 20 Internal Interrupt Sources, Including a High-priority, Low-latency Interrupt Request
  - 28 Programmable I/O Lines
  - 8-channel 11-bit Programmable Clock Prescaler Feeding the Timer, Watchdog, USARTs, SPIs
  - 3-channel 16-bit Timer/Counter
    5 Internal Clock Sources and 3 Configurable Sources (External Source or Cascaded Timer Configuration)
    2 Multi-purpose Output Pins plus 1 Output Dedicated to the ADDA Interface plus 3 Outputs Dedicated to the mAgic DSP
  - 2 USARTs
    2 Dedicated Peripheral Data Controller (PDC) Channels per USART
    1 USART Supporting Full Modem Interface
  - 2 Master/Slave SPI Interfaces
    2 Dedicated Peripheral Data Controller (PDC) Channels per SPI
    8- to 16-bit Programmable Data Length
    4 External Slave Chip Selects for each SPI
  - Programmable Watchdog Timer
  - ADDA (A/D and D/A Converters) Interface Supporting up to 4 Analog to Digital and 4 Digital to Analog, Stereo 24-bit Converters
  - IEEE 1149.1 JTAG Boundary Scan on all Active Pins
- Efficient ARM - DSP Interface Based on 1K x 40-bit Dual Ported Shared Memory, Memory Mapped Register Access, and Interrupt Lines
- 1.8 V Core Operating Voltage, 3.3 V I/O Operating Voltage
- On-chip PLL for 100 Mhz Operation from 25 Mhz Reference Clock
- 352-ball PBGA Package

DIOPSIS 740 Dual Core DSP

AT572D740

Summary

7001AS–DSP–03/04

Note: This is a summary document. A complete document is not available at this time. For more information, please contact your local Atmel sales office.

Description

DIOPSIS 740 is a Dual CPU Processor integrating a mAgic DSP and an ARM7TDMI™ RISC MCU, plus a total of 245 Kbytes SRAM. The system combines the flexibility of the ARM7TDMI RISC controller with the very high performance of the DSP.

mAgic is a high performance VLIW DSP delivering 1 Giga floating-point operations per second (GFLOPS) at a clock rate of 100 MHz. It has 512 data registers, 16 address registers, 10 independent operating units and 2 independent address generation units. For instance, activating all the computing units, it can produce one complete FFT butterfly per cycle. mAgic operates on 32-bit fixed-point and IEEE 754 40-bit extended precision floating-point numeric format. It has also on-chip 17K x 40-bit data memory locations and 8K x 128-bit program memory locations. Efficient usage of the internal program memory is achieved through a code compression mechanism.

An optimizing assembler frees the user from the burden of dealing with the parallelism of the processor resources and drastically simplifies the code development.

The ARM7TDMI™ embedded micro controller core is a member of the Advanced RISC Machines (ARM®) family of general purpose 32-bit microprocessors, which offer high performance and very low power consumption. The ARM architecture is based on Reduced Instruction Set Computer (RISC) principles, and the instruction set and the related decode mechanism are much simpler than those of micro programmed Complex Instruction Set Computers.

This simplicity results in a high instruction throughput and impressive real-time interrupt response. The ARM7TDMI™ supports 16-bit Thumb® subset of the most commonly used 32-bit instructions. These are expanded at run time with no degradation of system performance. This gives 16-bit code density (saving memory area and cost) coupled with 32-bit processor performance.

A rich set of peripheral and a 32 Kbytes internal memory provide a highly flexible and integrated system solution.

Pin Configuration

Table 1. D740 Ball Assignment (243 I/O)

Name	Ball	Name	Ball	Name	Ball	Name	Ball
ADDA_BRCK	C21	ARM_D[6]	W25	PIO[8]	AD23	SPI0_NSS[1]	A17
ADDA0_IN	B21	ARM_D[7]	Y24	PIO[9]	AE24	SPI0_NSS[2]	D17
ADDA1_IN	A22	ARM_D[8]	Y26	PIO[10]	AD22	SPI0_NSS[3]	B16
ADDA2_IN	C22	ARM_D[9]	Y25	PIO[11]	AC22	SPI0_SCK	D18
ADDA3_IN	D22	ARM_D[10]	AA26	PIO[12]	AE23	SPI1_MISO	B19
ADDA0_OUT	B22	ARM_D[11]	AA24	PIO[13]	AD21	SPI1_MOSI	A20
ADDA1_OUT	A23	ARM_D[12]	Y23	PIO[14]	AF22	SPI1_NSS	C18
ADDA2_OUT	C23	ARM_D[13]	AA25	PIO[15]	AE22	SPI1_NSS [1]	C19
ADDA3_OUT	B23	ARM_D[14]	AB26	PIO[16]	AD20	SPI1_NSS [2]	A18
ADDA_TOPLL	A24	ARM_D[15]	AB24	PIO[17]	AF21	SPI1_NSS [3]	B17
ADDA_WCK	B24	ARM_NCS0	H25	PIO[18]	AC20	SPI1_SCK	A19
ARM_A[0]	A25	ARM_NCS1	J26	PIO[19]	AE21	TEST_CLK (dnc)	M25
ARM_A[1]	D24	ARM_NCS2	K24	PIO[20]	AD19	USART0_RXD	AE17
ARM_A[2]	C25	ARM_NCS3	J25	PIO[21]	AF20	USART0_SCK	AF17
ARM_A[3]	E24	ARM_NRD	K23	PIO[22]	AC19	USART0_TXD	AE18
ARM_A[4]	D26	ARM_NWEB0	K26	PIO[23]	AE20	USART1_CTS	AD12
ARM_A[5]	D25	ARM_NWEB1	L24	PIO[24]	AD18	USART1_DCD	AE14
ARM_A[6]	F24	BIST_RES (dnc)	H1	PIO[25]	AE19	USART1_DSR	AC14
ARM_A[7]	E26	BIST_RUN (dnc)	H3	PIO[26]	AF18	USART1_DTR	AF14
ARM_A[8]	E25	FPU_EXC	AD15	PIO[27]	AD17	USART1_RI	AF15
ARM_A[9]	G24	FPU_HALT	AD13	PLL_CLKIN	N24	USART1_RTS	AF16
ARM_A[10]	F26	FPU_MODE	AE15	PLL_CLKOUT	N25	USART1_RXD	AC15
ARM_A[11]	G23	ICE_NTRST	K25	PLL_DIV (dnc)	P24	USART1_SCK	AD16
ARM_A[12]	F25	ICE_TCK	M23	PLL_DN (dnc)	T25	USART1_TXD	AC17
ARM_A[13]	H24	ICE_TDI	L26	PLL_EN	L25	XM_A[0]	AC12
ARM_A[14]	G26	ICE_TDO	N23	PLL_LFT	T24	XM_A[1]	AE13
ARM_A[15]	H23	ICE_TMS	M24	PLL_LOCK	R24	XM_A[2]	AD11
ARM_A[16]	G25	JCFG	M26	PLL_TST (dnc)	N26	XM_A[3]	AD10
ARM_A[17]	J24	PIO[0]	AB23	PLL_UP (dnc)	U23	XM_A[4]	AE11
ARM_A[18]	H26	PIO[1]	AB25	RESET	AD14	XM_A[5]	AC10
ARM_D[0]	V24	PIO[2]	AC26	SCAN_EN (dnc)	G2	XM_A[6]	AD9
ARM_D[1]	U25	PIO[3]	AC24	SCAN_TEST (dnc)	F1	XM_A[7]	AE10
ARM_D[2]	V26	PIO[4]	AC25	SINGLE	AE16	XM_A[8]	AF9
ARM_D[3]	V25	Notes: 1. PIO[5]	AD26	SPI0_MISO	C20	XM_A[9]	AE9

Table 1. D740 Ball Assignment (243 I/O) (Continued)

Name	Ball	Name	Ball	Name	Ball	Name	Ball
ARM_D[4]	W24	PIO[6]	AD25	SPI0_MOSI	B20	XM_A[10]	AD8
ARM_D[5]	V23	PIO[7]	AE26	SPI0_NSS	C17	XM_A[11]	AF8
XM_A[12]	AC9	XM_D[14]	U3	XM_D[39]	C14	XM_CLKOUT[0]	J4
XM_A[13]	AE8	XM_D[15]	V2	XM_D[40]	U4	XM_CLKOUT[1]	H2
XM_A[14]	AD7	XM_D[16]	L1	XM_D[41]	U1	XM_CLKOUT[2]	G1
XM_A[15]	AF7	XM_D[17]	K3	XM_D[42]	T3	XM_D[0]	AD2
XM_A[16]	AE7	XM_D[18]	L2	XM_D[43]	U2	XM_D[64]	B7
XM_A[17]	AF6	XM_D[19]	K4	XM_D[44]	R4	XM_D[65]	C9
XM_A[18]	AC7	XM_D[20]	K1	XM_D[45]	R3	XM_D[66]	A8
XM_A[19]	AE6	XM_D[21]	K2	XM_D[46]	T2	XM_D[67]	A9
XM_A[20]	AF5	XM_D[22]	J1	XM_D[47]	R1	XM_D[68]	C10
XM_A[21]	AD5	XM_D[23]	J2	XM_D[48]	P3	XM_D[69]	B9
XM_A[22]	AC5	XM_D[24]	E3	XM_D[49]	R2	XM_D[70]	D10
XM_A[23]	AE5	XM_D[25]	E4	XM_D[50]	N3	XM_D[71]	A10
XM_D[1]	AB3	XM_D[26]	E2	XM_D[51]	P1	XM_D[72]	A13
XM_D[2]	AC1	XM_D[27]	D1	XM_D[52]	N1	XM_D[73]	B13
XM_D[3]	AA3	XM_D[28]	D3	XM_D[53]	M4	XM_D[74]	A14
XM_D[4]	AB1	XM_D[29]	D2	XM_D[54]	N2	XM_D[75]	D15
XM_D[5]	AB2	XM_D[30]	C1	XM_D[55]	M2	XM_D[76]	B14
XM_D[6]	AA1	XM_D[31]	D5	XM_D[56]	C6	XM_D[77]	A15
XM_D[7]	Y4	XM_D[32]	C11	XM_D[57]	A5	XM_D[78]	B15
XM_D[8]	AA2	XM_D[33]	D12	XM_D[58]	C7	XM_D[79]	A16
XM_D[9]	Y1	XM_D[34]	A11	XM_D[59]	A6	XM_GNT	F2
XM_D[10]	W4	XM_D[35]	C12	XM_D[60]	D7	XM_NCS	E1
XM_D[11]	Y2	XM_D[36]	B11	XM_D[61]	C8	XM_NWE	F3
XM_D[12]	W1	XM_D[37]	A12	XM_D[62]	A7	XM_REQ	G4
XM_D[13]	V1	XM_D[38]	C13	XM_D[63]	D8		

Note: dnc = do not connect pins. These pins are reserved for test use only and are not described in Table 6.

Table 2. D740 Ball Assignment (VDD = 3.3V)

D6	F4	L4	AC6	D11	F23	L23	AC11	D16	AA4
T4	D21	AC16	AA23	T23	AC21				

Table 3. D740 Ball Assignment (VDDI = 1.8V)

B18	B12	B6	T1	W3	AD6	AF11	AF19	AF23	W26
E23									

Table 4. D740 Ball Assignment (VDDPLL = 1.8V)

P25	R26								

Table 5. D740 Ball Assignment (GND)

A1	C3	D23	W23	AD3	AF25	A2	C24	AC4	AD24
H4	AF26	A26	D4	AC8	J23	AE1	B2	D9	N4
AC13	AE2	B25	AE25	AC18	P23	D14	B26	AF1	AC23
D19	V4								

All balls not comprised in Tables 1 to 5 are “not connected”.

Pin name conventions

Pin names are built using the following structure:

(functional block name) _ (activity level) (line name) (bus index)

where:

- functional block name = name of the functional block to which the pin belongs
- activity level = “n” for low active lines; blank for high active lines
- line name = name of the function of the pin line
- bus index = number (in [ ]) corresponding to the index when the pin line is an element of a bus

Pin Description

Table 6. D740 Pin Description

Module	Name	Function	Type	Active Level	Notes
ADDA	ADDA_BRCK	ADDA Bit rate clock	in		digital serial audio stream bit rate clock (64 x F sampling)
ADDA	ADDA0_IN	ADDA 0 input channel	in		24 bit Left + 24 bit right digital serial stereo audio stream
ADDA	ADDA1_IN	ADDA 1 input channel	in		24 bit Left + 24 bit right digital serial audio stream
ADDA	ADDA2_IN	ADDA 2 input channel	in		24 bit Left + 24 bit right digital serial audio stream
ADDA	ADDA3_IN	ADDA 3 input channel	in		24 bit Left + 24 bit right digital serial audio stream
ADDA	ADDA0_OUT	ADDA 0 output channel	in		24 bit Left + 24 bit right digital serial stereo audio stream
ADDA	ADDA1_OUT	ADDA 1 output channel	in		24 bit Left + 24 bit right digital serial audio stream
ADDA	ADDA2_ OUT	ADDA 2 output channel	in		24 bit Left + 24 bit right digital serial audio stream
ADDA	ADDA3_ OUT	ADDA 3 output channel	out-02		24 bit Left + 24 bit right digital serial audio stream
ADDA	ADDA_TOPLL	ADDA clock generator Strobe	out-02		F Sampling toward an external PLL for ADCs/DACs synchronism generation
ADDA	ADDA_WCK	ADDA Word clock	out-03		F Sampling clock toward ADCs/DACs
ARM	ARM_A[18:0]	ARM external memory address bus	out-02		
ARM	ARM_D[15:0]	ARM external memory data bus	bi-02		(internal Pull-Down)
ARM	ARM_NCS0	ARM external memory Chip select command 0	out-02	low	
ARM	ARM_NCS1	ARM external memory Chip select command 1	out-02	low	
ARM	ARM_NCS2	ARM external memory Chip select command 2	out-02	low	
ARM	ARM_NCS3	ARM external memory Chip select command 3	out-02	low	
ARM	ARM_NRD	ARM external Memory Read enable	bi-02	low	
ARM	ARM_NWEB0	ARM external memory Low Byte Write enable	bi-03	low	data byte d[7:0]
ARM	ARM_NWEB1	ARM external memory High Byte Write enable	bi-03	low	data byte d[15:8]
mAgic	FPU_HALT	ARM Fast IRQ from mAgic “halt”	out-02	high	To be used for monitoring

Table 6. D740 Pin Description (Continued)

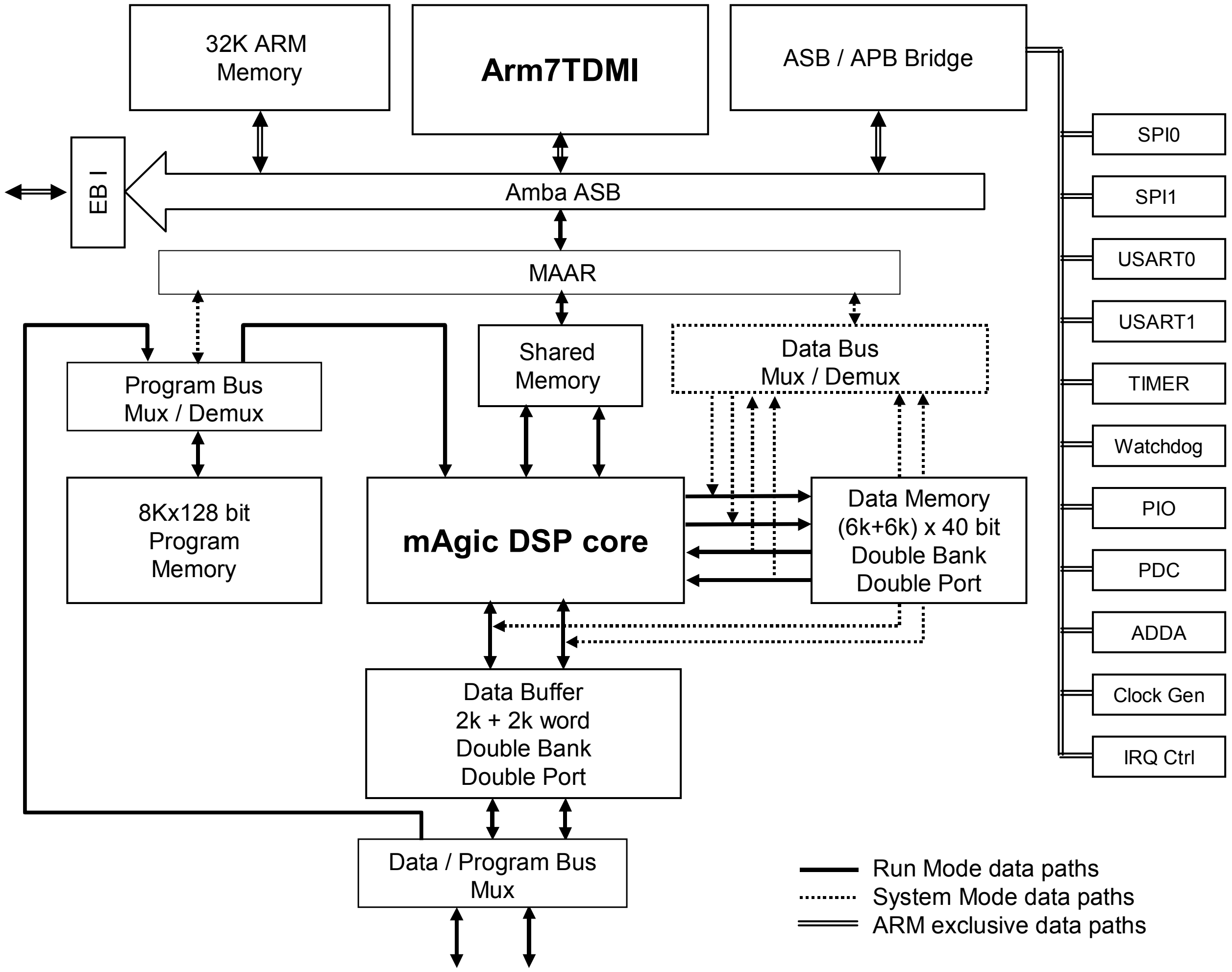
Module	Name	Function	Type	Active Level	Notes
mAgic	FPU_EXC	ARM IRQ15 from mAgic “exception”	out-02	high	To be used for monitoring
mAgic	FPU_MODE	ARM IRQ25 from mAgic “mode”	out-02		To be used for monitoring 0 = mAgic in system mode 1 = mAgic in run mode
JTAG	ICE_NTRST	JTAG Test reset	in	low	(internal Pull-Up)
JTAG	ICE_TCK	JTAG Test clock	in		
JTAG	ICE_TDI	JTAG Test data input	in		(internal Pull-Up)
JTAG	ICE_TDO	JTAG Test data output	out-02		
JTAG	ICE_TMS	JTAG Test mode	in	low	(internal Pull-Up)
D740	JCFG	ARM JTAG / D740 Boundary Scan selection	in		0→ D740 Boundary Scan 1→ ARM JTAG
PIO	PIO[27:0]	Parallel Input/Output	bi-02		general purpose programmable I/Os or ARM peripheral I/Os
PLL	PLL_CLKIN	Reference clock	in		25MHz (max) if PLL_EN =1 100MHz (max) if PLL_EN =0
PLL	PLL_CLKOUT	PLL Clock output	out-02		100MHz (max) if PLL_EN =1 fixed low if PLL_EN = 0
PLL	PLL_EN	Pll enable (PLL_CLKIN x4 multiply)	in	high	1 → system clock = PLL_CLKIN x 4 0 → system clock = PLL_CLKIN
PLL	PLL_LFT	PLL lowpass filter input	in		
PLL	PLL_LOCK	PLL lock condition	out-02	high	To be used for monitoring
D740	RESET	System reset	in	low	asynchronous
mAgic	SINGLE	Single user on mAgic external memory	in	high	(internal Pull-Up) 0 → Not default user of shared XM 1 → Single user of not shared XM or default user of shared XM
SPI	SPI0_MOSI	SPI 0 Master Out/Slave In data	bi-02		SPI SLV → data input SPI MST → data output
SPI	SPI0_MISO	SPI 0 Master In/Slave Out data	bi-02		SPI SLV → data output SPI MST → data input
SPI	SPI0_NSS	SPI 0 Input/Output Chip select	bi-02		SPI SLV → CS Input SPI MST → CS 0 Output
SPI	SPI0_NSS[3:1]	SPI 0 Output Chip Selects	out-02		SPI SLV → n.a. SPI MST → CS 3, 2, 1 Outputs
SPI	SPI0_SCK	SPI 0 Serial clock	bi-03		SPI SLV → clock input SPI MST → clock output
SPI	SPI1_MOSI	SPI 1 Master Out/Slave In data	bi-02		SPI SLV → data input SPI MST → data output

Table 6. D740 Pin Description (Continued)

Module	Name	Function	Type	Active Level	Notes
SPI	SPI1_MISO	SPI 1 Master In/Slave Out data	bi-02		SPI SLV → data output SPI MST → data input
SPI	SPI1_NSS	SPI 1 Input/Output Chip select	bi-02		SPI SLV → CS Input SPI MST → CS 0 Output
SPI	SPI1_NSS[3:1]	SPI 1 Output Chip Selects	out-02		SPI SLV → n.a. SPI MST → CS 3, 2, 1 Outputs
SPI	SPI1_SCK	SPI 1 Serial clock	bi-03		SPI SLV → clock input SPI MST → clock output
USART	USART0_RXD	USART 0 Data in	in		(internal Pull-Down)
USART	USART0_SCK	USART 0 Serial clock	bi-03		for synchronous mode only
USART	USART0_TXD	USART 0 Data out	bi-02		used as output
USART	USART1_CTS	USART 1 Clear to send	in		
USART	USART1_DCD	USART 1 Data carriage detect	in		
USART	USART1_DSR	USART 1 Data set ready	in		
USART	USART1_DTR	USART 1 Data terminal ready	out-02		
USART	USART1_RI	USART 1 Ring indicator	in		
USART	USART1_RTS	USART 1 Request to send	out-02		
USART	USART1_RXD	USART 1 Data in	in		(internal Pull-Down)
USART	USART1_SCK	USART 1 Serial clock	bi-03		for synchronous mode only
USART	USART1_TXD	USART 1 Data out	bi-02		used as output
mAgic	XM_A[23:0]	mAgic external Memory address bus	out-03		
mAgic	XM_CLKOUT[2:0]	mAgic external Memory clocks	out-03		100MHz (max) One line for up to three mAgic XM chip.
mAgic	XM_D[39:0]	mAgic external Memory data bus	bi-03		Right bank (internal Pull-Down)
mAgic	XM_D[79:40]	mAgic external Memory data bus	bi-03		Left bank (internal Pull-Down)
mAgic	XM_GNT	mAgic shared external memory bus grant	out-02	high	
mAgic	XM_NCS	mAgic external Memory Chip select	out-03	low	
mAgic	XM_NWE	mAgic external Memory Write enable	out-03	low	
Power	VDD	IO power supply	Power		3.3 nominal Supply
Power	VDDI	Core power supply	Power		1.8 nominal Supply
Power	VDDPLL	PLL power supply	Power		1.8 nominal Supply
Ground	GND	D740 ground reference	Ground		common to all Supplies

Block Diagram

Figure 1. D740 Architecture



Architectural Overview

DIOPSIS 740 (also named D740) is a high performance dual-core processing platform for audio, communication and beam-forming applications, integrating a floating-point DSP (mAgic DSP) and an ARM7TDMI™ Reduced Instruction Set Computer (RISC). The D740 is optimally suited for floating point applications with a significant need for complex domain computations like FFT and frequency domain phase-shift algorithms, requiring high dynamic range and maximum numerical precision.

The D740 combines the flexibility of the ARM7 RISC controller with the very high performance of the DSP oriented VLIW architecture of mAgic.

System management

The availability of a standard RISC on-chip lowers software development effort for non critical and control segments of the application. ARM7TDMI supports the usage of light RTOS and has efficient interrupt management, leaving mAgic fully available for the numerically intensive part of the application. The synchronization between the two processors can be either based on software polling on semaphores or on interrupts.

The ARM is the D740 master processor. The bootstrap sequence of the D740 starts from the bootstrap of the ARM from its external non-volatile memory. The ARM then boots mAgic from a non-volatile memory. After bootstrap the D740 can start its normal operations. The DSP side of many applications can be implemented on the D740 using only the internal memory. In fact the program memory size of 8K by 128-bit coupled with the availability of the code compression, gives an equivalent on-chip program memory size of about 24K instructions (typical).

The ARM standard In-Circuit Emulation debug interface is supported via the ICE port.

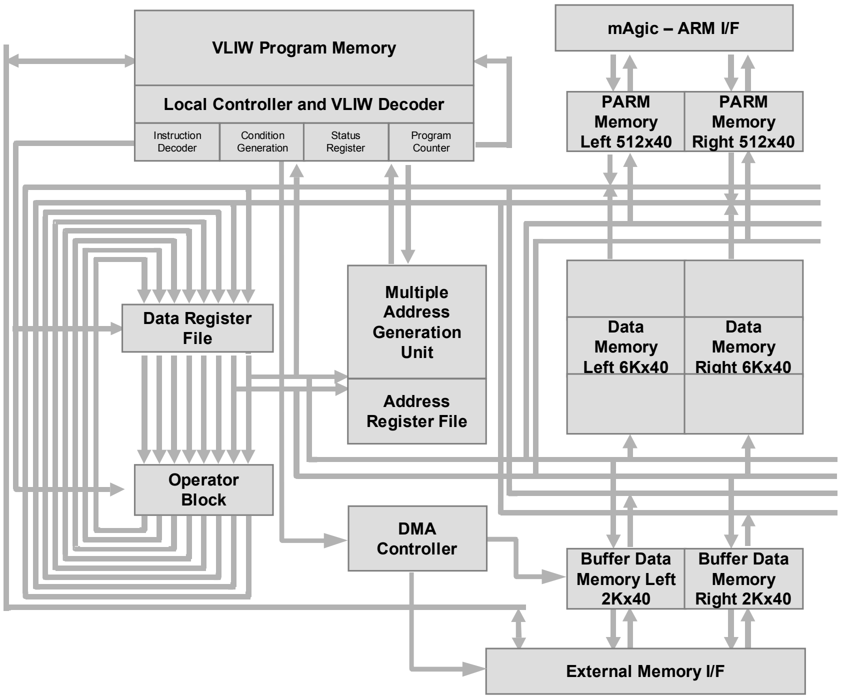
mAgic DSP Processor

The mAgic DSP is the VLIW numeric processor of the D740. It operates on IEEE 754 40-bit extended precision floating-point and 32-bit integer numeric format. The main components of the DSP subsystem are the core processor, the on-chip memories and the interfaces to and from the ARM subsystem. The operators block, the register file, the address generation unit and the program decoding and sequencing unit compose the core processor. A short description of each block is given in the following paragraphs.

Core processor

mAgic is a VLIW engine, but from an user point of view, it works like a RISC machine by implementing triadic computing operations on data coming from the register file, and data move operations between the local memories and the register file. The operators are pipelined for maximum performance. The pipeline depth depends on the operator used. The operations scheduling and parallelism are automatically defined and managed at compile time by the assembler-optimizer, allowing efficient code execution. In order to give the best support to the RISC-like programming model, mAgic is equipped with a complex 256-entry register file. It can be used as a complex register file (real + imaginary part), or as a dual register file for vectorial operations. When performing single instructions the register file can be used as an ordinary 512 register file. Both the left and right side of the register file are 8-ported, making a total of 16 I/O port available for the data move to and from the operator block and the memory. The total data bandwidth between the register file and the operator block is 70 bytes per clock cycle, avoiding bottlenecks in the data flow between the two units.

Figure 2. mAgic DSP Block Diagram



The operators block, the register file, the address generation unit and the program-sequencing unit compose the core processor. The Operators Block contains the hardware that performs arithmetical operations. It works on 32-bit integers and IEEE 754 extended precision 40-bit floating-point data.

The Operators Block is composed of four integer/floating point multipliers, an adder, a subtractor and two add-subtract integer/floating point units; moreover, it has two shift/logic units, a Min/Max operator and two seed generators for efficient division and inverse square root computation. The operators block is arranged in order to natively support complex arithmetic (single cycle complex multiply or multiply and add), fast FFT (single cycle butterfly computation) and vectorial computations. The peak performance of mAgic is achieved during single cycle FFT butterfly execution, when mAgic delivers 10 floating-point operations per clock cycle.

mAgic is equipped with two independent address generation units. It is able to generate up to two pairs of addresses, one to access the left and the right memory for reading and one to access the left and the right memory for writing. It is also used in the loop control to test if the end of a loop is reached. The Multiple Address Generation Unit (MAGU) supports linear addressing with stride, circular addressing and bit reversed addressing. The address generation unit has 16 registers.

The Program Address Generation Unit is devoted to control the correct Program Counter generation according to the program flow. It generates addresses for linear code execution as well as for non-sequential program flow. The Condition Generation Unit combines the flags generated by the operators to produce complex conditions flags used to control the program execution. Predicated instruction execution is supported for different groups of instructions: arithmetical instructions, memory write, immediate load, or all of them. The Program Address Generation Unit also allows to perform conditioned and unconditioned branch instructions, loops, call to subroutines and return from subroutines.

Internal memories, External memories and DMA

mAgic has four on-chip memory blocks: the Program Memory, the Data Memory, the Data Buffer, and the dual ported memory shared with the ARM processor.

An External Memory Interface multiplexes the Data accesses and the Program accesses to and from the External Memory.

The Program Memory stores the VLIW program to be executed by mAgic. It is 8K words by 128-bit single port memory. When mAgic is in System Mode the ARM can modify the content of the mAgic Program Memory in two different ways. The ARM can directly write a Program Memory location by accessing the memory address space assigned to the mAgic Program Memory in the ARM memory map. In this access mode the ARM writes four 32-bit words to four consecutive addresses at correct address boundaries, in order to properly complete a single VLIW word write cycle. The ARM can also modify the content of the mAgic Program Memory by initiating a DMA transfer from the External Memory to the mAgic Program Memory. In this access mode a single VLIW word is transferred from the mAgic External Memory to the mAgic Program Memory 64-bit per cycle, that is a complete word every two clock cycles. Due to the program compression scheme used, which allows an average program compression between 2 and 3, the code accessing capability of mAgic from its External Memory is greater than an instruction per clock cycle. When mAgic is in Run Mode, the ARM cannot get access to the mAgic Program Memory. When in Run Mode mAgic can initiate a DMA transfer from the External Memory to the mAgic Program Memory to load a new code segment.

The mAgic internal Data Memory is made of three memory pages, 2K words by 40-bit for the left data memory and 2K words by 40-bit for the right data memory, giving a total of 6K words for the left and for the right memory banks (a total of 12K words). Each Data Memory bank is a dual port memory that allows four simultaneous accesses, two read and two write. The core can access vectorial and single data stored in the Data Memory. Accessing complex data is equivalent to accessing vectorial data. During simultaneous read and write memory accesses, the MAGU generates two independent read and write addresses common to both the left and the right memory banks. The total available bandwidth between the Register File and the Data Memory is 20 bytes per clock cycle, allowing full speed implementation of numerically intensive algorithms (e.g. complex FFT and FIR).

The Buffer Memory is 2K words by 40-bit for both the left and the right memory. The Buffer Memory is a dual port memory. A port is connected to the core processor. The MAGU generates the Buffer Memory addresses for transferring data to and from the core. The second port of the Buffer Memory is connected to the External Memory Interface. The Buffer Memory does not support dual read and write accesses neither from the core nor from the External Memory Interface. The available bandwidth between the core processor and the Buffer Memory is equal to the available bandwidth between the External Memory Interface and the Buffer Memory: 10 bytes per clock cycle. The maximum External Memory size of mAgic is 16 Mword Left and Right (equivalent to 32 Mword or 160 Mbytes; 24-bit address bus). A DMA controller manages the data transfer between the External Memory and the Buffer Memory. The DMA controller can generate accesses with stride for the External Memory. The DMA transfers to and from the Buffer Memory can be executed in parallel with the full speed core instructions execution with zero-overhead and without the intervention of the core processor, except for initiating it.

The last memory block in the address space of the mAgic DSP is the memory shared (PARM) between mAgic and the ARM processor. It is a dual port memory 512 words by 40- bit for both the left and the right bank (total 1K by 40-bit). This memory can be used to efficiently transfer data between the two processors. The available bandwidth between the core processor and the shared memory is 10 bytes per clock cycle. On the

ARM side the available bandwidth is limited by the bus size of the ARM processor (32 bits) giving a bandwidth of 4 bytes per ARM clock cycle.

ARM interface (mAAr)

The D740 master is the ARM7 RISC processor. mAgic behaves as a standard AMBA ASB slave device, allowing access to different resources depending on the operating mode (Run or System).

In System Mode, mAgic halts its execution and the ARM takes control of it. When mAgic is in System mode the ARM can access many mAgic internal devices. The ability of the ARM to access internal mAgic resources in System Mode can be used for initialization and debugging purposes. By accessing the Command Register, the ARM can change the operating status of the DSP (Run/System Mode), initiate DMA transactions, force single or multiple step execution, or simply read the DSP operating status.

In Run Mode, mAgic works under direct control of its own VLIW program and the ARM has access only to the 1K x 40-bit dual ported shared memory (PARM) and to the mAgic Command Register.

In order to allow a tight coupling between the operations of mAgic and the ARM at run time, they can exchange synchronization signals, based on interrupts.

ARM System: ARM7TDMI Processor and Peripherals

The ARM7TDMI is a 32-bit RISC microprocessor; it is a member of the Advanced RISC Machines (ARM) family of general-purpose 32-bit microprocessors, offering high performance and very low power consumption.

The ARM architecture is based on Reduced Instruction Set Computer (RISC) principles, and the instruction set and related decode mechanism are much simpler than those of microprogrammed Complex Instruction Set Computers. This simplicity results in a high instruction throughput and a real-time interrupt response. Pipelining is employed so that all parts of the processing and memory systems can operate continuously. The typical operating scheme of the ARM7TDMI is the sequence fetch-decode-execute.

The ARM7TDMI processor employs the architectural strategy known as THUMB. THUMB instructions operate with the standard ARM register configuration, allowing excellent interoperability between ARM and THUMB states. Each 16-bit THUMB instruction has a corresponding 32-bit ARM instruction with the same effect on the processor model. The 16-bit instructions are expanded at run time with no degradation of the system performance. This provides far better performance than a 16-bit architecture, with better code density than a 32-bit architecture.

The ARM7TDMI processor is built around a bank of 37 32-bit registers and six status registers. The ARM7TDMI supports seven operation modes:

1. User (usr): The normal ARM program execution state
2. FIQ (fiq): Fast Interrupt reQuest; it is connected to the mAgic Halt signal
3. IRQ (irq): Used for general-purpose interrupt handling
4. Supervisor (svc): Protected mode for the operating system
5. Abort mode (abt): Entered after data or instruction prefetch abort
6. System (sys): A privileged user mode for the operating system
7. Undefined (und):Entered when an undefined instruction is executed

Mode changes can be made under software control or can be brought about by external interrupts or exception processing. Most application programs execute in User mode. The non-user modes - known as privileged modes – are entered in order to service interrupts or exceptions, or to access protected resources. Each operating mode has dedicated banked registers for fast exception handling. The FIQ mode has five additional banked working registers, r8_fiq to r12_fiq, to enhance interrupt processing speed.

The ARM7TDMI processor operates in little-endian mode.

To speed-up critical routine execution or critical data segment access, the ARM7 is equipped with 32 Kbyte of zero wait states on-chip memory.

The ARM system has two buses. The main bus is the ASB (ARM System Bus). The APB (ARM Peripheral Bus) is designed for accesses to on-chip peripherals. The AMBA Bridge provides an interface between the ASB and the APB.

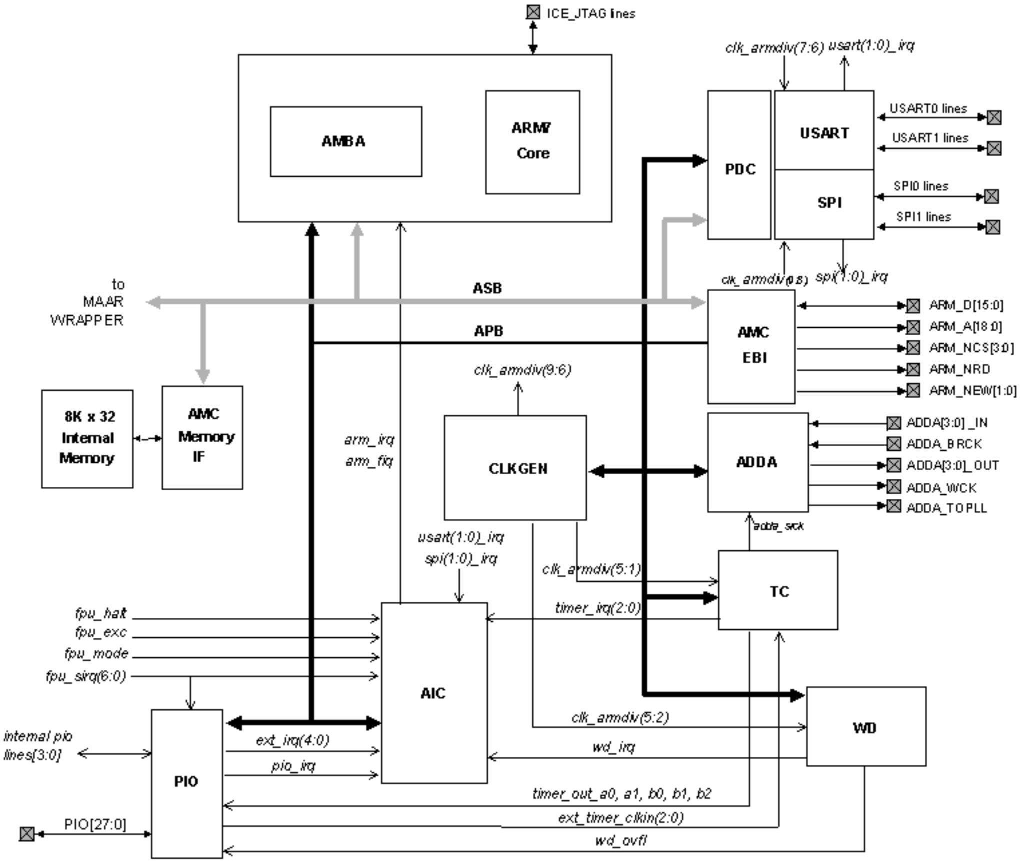
The D740 is equipped with a set of peripherals controlled by the ARM. An on-chip Peripheral Data Controller (PDC) transfers data between the on-chip USARTs/SPI and the on- and off-chip memories in the DMA without the intervention of the processor. Most importantly, the PDC removes the processor interrupt handling overhead and significantly reduces the number of clock cycles required for data transfer.

Each peripheral has a 16K-byte address space allocated in the upper 3M bytes of the 4Gbyte address space. The peripheral register set is composed of control, mode, data, status, and interrupt registers. To maximize the efficiency of bit manipulation, frequently written registers are mapped into three memory locations.

A short description of the available peripherals is given in the following.

- **EBI** (External Bus Interface): the EBI generates the signals that control the access to the External Memory or peripheral devices.
- **ADDA** (Analog to Digital and Digital to Analog interface): the ADDA provides 4 channel serial interface toward stereo audio 24-bit ADC and DAC.
- **PDC** (Peripheral Data Controller): The PDC provides 8 communication channels dedicated to the two USARTs and to the two SPIs. One PDC channel is connected to the receiving channel and the one to the transmitting channel of each peripheral.
- **USART** (Universal Synchronous / Asynchronous Receiver / Transmitter): two, full-duplex, universal synchronous/asynchronous receiver/transmitters provide a simple standard communication way managed by the Peripheral Data Controller.
- **SPI** (Serial Peripheral Interface): two four-wire serial interfaces provide a simple industry-standard communication way managed by the Peripheral Data Controller.
- **AIC** (Advanced Interrupt Controller): the AIC is an 8-level priority, individually-maskable, vectored interrupt controller. The interrupt controller is connected to the NFIQ (fast interrupt request) and the NIRQ (standard interrupt request) inputs of the ARM7TDMI processor.
- **PIO** (Parallel I/O Controller): The PIO features 32 programmable I/O lines, 28 PIO lines are available on D740 pads, while the remaining 4 are only internal.
- **TC** (Timer Counter): the TC contains three identical 16-bit timer/counter channels.
- **WD** (Watchdog Timer): the WD can be used to guard against system lock-up if the software becomes trapped in a deadlock. If an overflow occurs, the watchdog timer generates processor interrupts via the Advanced Interrupt Controller (AIC) and an external low pulse through the PIO.
- **CLKGEN** (Clock Generator): The clock generator provides divided clocks for several peripherals: the Timer Counter, the Watchdog, the USARTs and the SPIs.

Figure 3. Armsystem Architecture



Development Tools

D740 is supported with a complete set of software and hardware development tools.

MADE

The D740 is supported by a set of development tools integrated into a visual development environment called MADE (Multicore Application Development Environment).

MADE provides the user with an integrated environment for producing applications for both the D740 cores, the ARM7TDMI and the mAgic DSP, by means of a common project management and support for the MARMOS Minimal Bios.

Code generation tools for the ARM include the GNU Code Development Chain for ARM7 (C-C++ compiler, assembler, linker and utilities) and the ARM SDT Code Development Chain (C-C++ compiler, assembler, linker and utilities).

Code generation tools for mAgic include C compiler (GNU gcc based, ANSI compliant), VLIW assembler-optimizer, code compressor, linker and utilities.

MADE supports the MARMOS Minimal Bios, a set of helper functions for the ARM-mAgic intercommunication and the D740 peripherals management. MARMOS gives the user the basic APIs for building an integrated ARM-mAgic application.

MADE provides the user with a simulation engine and an emulation kernel: the Cycle-Accurate simulator and the D740 emulator board support.

JTAG-ICE

The ARM Standard In-Circuit-Emulation debug interface is supported via the JTAG-ICE port of the D740.

When the ARM ICE configuration is selected, the usual debug capabilities for the ARM System are supported, while the support for the mAgic core is limited to memory and status registers inspection.

The 5 jtag pins are shared between ARM7TDMI ICE functionality and the DIOPSIS 740 chip Boundary Scan Logic. The “JCFG” pin acts as ARM jtag / D740 BSL selector. When “JCFG” pin is high the ARM ICE is selected, while DIOPSIS 740 BSL is selected when “JCFG” is low.

JTST

JTST is a low cost general-purpose module that provides the appropriate resources in order to test DIOPSIS 740. JTST provides the following resources to DIOPSIS 740:

- mAgic SSRAM, ARM FLASH and SRAM
- 4 Stereo Audio 20 bit CODECs
- 1 USB 2.0 Full (12 Mbps)
- 2 RS232/LVTTL a/synchronous serial I/O lines
- 2 SPI serial I/O lines
- Reset Logic (Power ON, Push Button, WDG)
- IO connectors (USART, SPI, USB, PIO, AUDIO)
- PLL-Clock Logic (25 MHz oscillator + CLK connector)
- DIP SWITCH & Status 7-segment Display
- Voltage Regulators 5V/3.3V & 5V/1.8V
- M-ICE JTAG

Mechanical Drawing

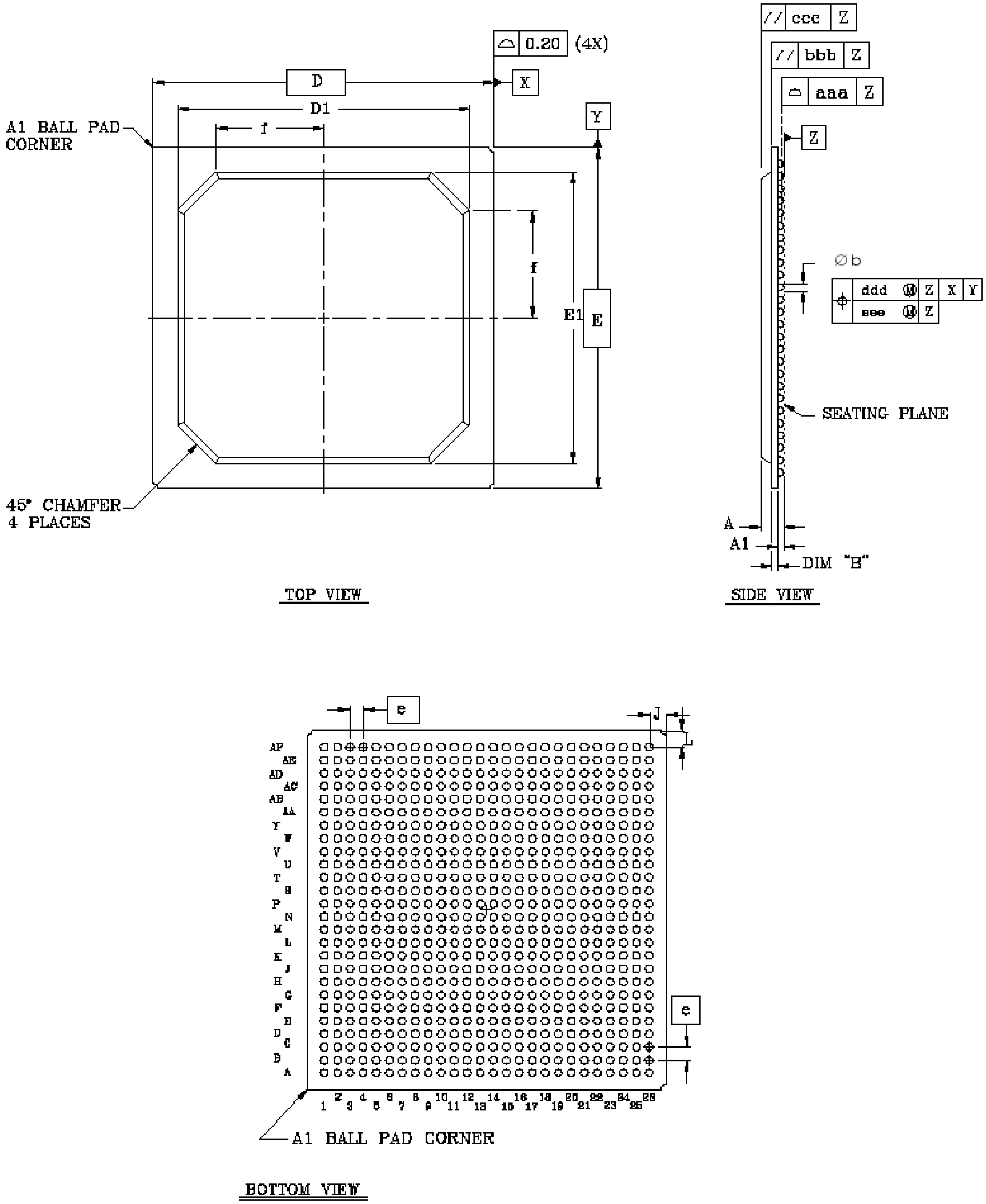


Table 7. D740 Dimensions (mm)

Symbol	Min	Nom	Max
A1	0.50	0.60	0.70
∅ b	0.60	0.75	0.90
aaa		0.30	
bbb		0.25	
ccc		0.35	
ddd		0.30	
eee		0.15	
A	2.12	2.33	2.56
Dim “B”	0.44	0.52	0.60
e REF		1.27	
D/E	34.8	35.0	35.2
D1/E1		30.0	30.7
f REF		11.0	
J/L REF		1.62	

Power Dissipation

The D740 has three kinds of power supply pins:

- VDDCORE pins, which power the chip core (1.8V)
- VDDIO pins, which power the I/O lines (3.3V)
- VDDPLL pins, which power the oscillator and PLL cells (1.8V)

The total power dissipation is the sum of two basic contributions:

$$P_D = P_{IO} + P_{CORE}$$

P_{IO} represents the contribute due to the IO pads current and the output load current.

P_{CORE} represents the contribute due to the internal activity current.

The following table defines the current consumption on different conditions:

Table 8. Power Dissipation

Parameters	typical conditions		worst conditions	
	Idd IO (3.3V) mA	**Idd CORE (1.8V) mA**	**Idd IO (3.3V) mA**	**Idd CORE (1.8V) mA**
Idd $_{peak}$	330	460	425	600
Idd $_{high}$	120	400	155	520
Idd $_{no\ ext}$	25	390	35	500
Idd $_{sys\ mode}$	25	100	35	135
Idd $_{rst}$	10	160	15	205

- Idd $_{peak}$ = mAgic FFT; both mAgic and ARM ext mem written 100% with continuous toggling data
- Idd $_{high}$ = mAgic FFT; both mAgic and ARM ext mem read and written alternatively 100% with 50% toggling data
- Idd $_{no\ ext}$ = mAgic FFT; ARM FLASH access 100%; no mAgic ext mem access
- Idd $_{sys\ mode}$ = mAgic in system mode; ARM FLASH accesses 100%;
- Idd $_{rst}$ = D740 under reset
- typical condition = typical process; Tj = 25°; Vdd = nom
- worst condition = worst process; Tj = 100°; Vdd = nom + 10%

To estimate power consumption for a specific application use the following equation where % is the amount of time your program spends in that state and each “Idd” contribute corresponds to “IO” or “CORE” columns:

$$P_{CORE} = ((\%_{peak} \times Idd_{peak}) + (\%_{high} \times Idd_{high}) + (\%_{no\ ext} \times Idd_{no\ ext}) + (\%_{sys\ mode} \times Idd_{sys\ mode}) + (\%_{rst} \times Idd_{rst})) \text{ x } 1.8$$

$$P_{IO} = ((\%_{peak} \times Idd_{peak}) + (\%_{high} \times Idd_{high}) + (\%_{no\ ext} \times Idd_{no\ ext}) + (\%_{sys\ mode} \times Idd_{sys\ mode}) + (\%_{rst} \times Idd_{rst})) \text{ x } 3.3$$

Note: Idd $_{peak}$ represents worst-case processor operation (for Idd IO particularly) and it is not considerable for also for hard applications where all data bits do not toggle every cycle.

Reliability Data

The following table summarizes some basic data that can be used in reliability calculations.

Table 9. Silicon Block Size

Parameters	Data	Unit	Data	Unit
Logic Gates	585	Kgates	10.5	mm^2
Memories	12	M transistors	18	mm^2
Register File	0.3	M transistors	5.1	mm^2
total Device Die Size (pad excluded)			45	mm^2

Ordering Guide

Table 10. Ordering Information

Part Number	Temperature Range	Working Frequency	Operating Supplies	Package
AT572D740	0°C - 70°C	100 MHz	3.3V (I/O) & 1.8V (core)	352PBGA

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe

Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia

Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan

9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards

Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive

Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/ High Speed Converters/RF Datacom

Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests

www.atmel.com/literature

Disclaimer: Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

© Atmel Corporation 2003. **All rights reserved.** Atmel® and combinations thereof, aaa®, bbb® and ccc® are the registered trademarks, and aaa™, bbb™ and ccc™ are the trademarks of Atmel Corporation or its subsidiaries. aaa®, bbb® and ccc® are the registered trademarks, and aaa™, bbb™ and ccc™ are the trademarks of xxxx Company. Other terms and product names may be the trademarks of others.

Printed on recycled paper.