BAZIL: A Multi-Core Architecture for Flexible Broadband Processing

Neal S. Stollon, Ph.D, P.E.
LSI Logic Corporation

Björn Sihlbom, M.Sc.E.E.
Ericsson Microwave Systems AB

Abstract:
The combination of digital signal processing and programmable logic provides a robust platform for performing a range of communications applications, that require significant levels of both arithmetic and bit level pre- and post-processing. This paper discusses a heterogeneous signal processing architecture and demonstration silicon code named BAZIL (Broadband Architecture based on ZSP400 and Integrated programmable Logic cores). The paper focuses on BAZIL architecture, including; functionality, structure, integration, and connectivity of LSI Logic's embedded DSP (ZSP) and programmable logic (ePLC) cores in this signal processing platform.

Introduction:
A combination of signal and logical processing operations are common in a range of applications in the areas of wireless, imaging, and broadband communications processing. Architectures suited to one type of processing are typically not suited or appropriate for the other. General-purpose architectures are limited both in flexibility and efficiency for DSP operations. DSP architectures, developed for arithmetic operations, are not optimal in functions with extensive bit level manipulations. Heterogeneous architectures, that incorporate different processing cores, provide one solution to this tradeoff. Further complicating processing performance tradeoffs, algorithms and standards in many (emerging) areas of signal processing, most notably in communications, are evolving. This diversity of processing and need for flexibility and reconfigurability of operation in all but the most rigid of applications make fully programmable solutions attractive to system designers. [1] The BAZIL architecture addresses these issues by integrating DSP (LSI's ZSP400 core) and programmable logic (ePLC core developed by Adaptive Silicon) resources to create a fully programmable communications platform to handle both types of processing. This paper addresses BAZIL's architectural features, its constituent cores, architectural issues associated with this type of architecture, and approaches to chip level integration such as on-chip buses and peripherals. The paper concludes with preliminary information on metrics of BAZIL silicon.

Architecture:
BAZIL is based on the basic concept of intra-chip based DSP and programmable logic block co-processing. Key blocks in BAZIL are LSI Logic ZSP400 (ZSP) and ePLC cores, which are integrated with other LSI Logic peripherals for external memory and data access. The block diagram in Figure 1 shows the processing blocks (ZSP, ePLC subsystems) along with supporting external controllers and interface logic to form the BAZIL architecture. BAZIL's heterogeneous architecture can be controlled by bus masters, which include the ZSP and DMA controller, and AHB bridge. These alternative methods of on-chip control of cores and peripherals allows for robust alternatives in system programming.

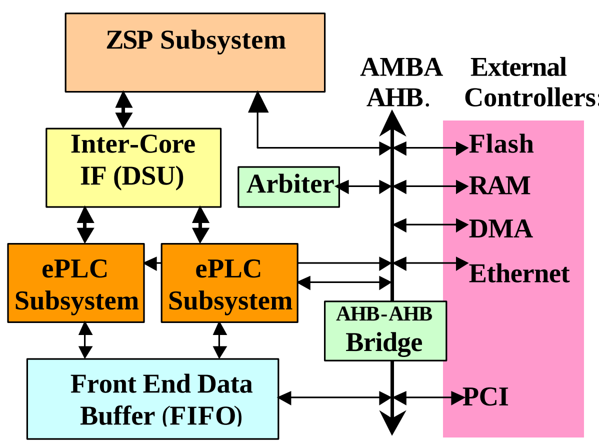

Figure 1. BAZIL Architectural Block Diagram

BAZIL supports combined boolean and DSP processing through a flexible core interconnect scheme. There are two types of inter-core communication supported in BAZIL architecture. An AMBA High Speed Bus (AHB) provides an arbitrated bus for inter-core communication As AMBA is a strategic on-chip bus architecture within LSI Logic, both ZSP and ePLC cores support AHB interfaces. [2, 3] The DSP and ePLC blocks are additionally interconnected through a higher bandwidth inter-core interface (DSU) that allows higher throughput and data sharing between cores. Both types of intra-chip communications are addressed in the Inter-Core Communication section. Since BAZIL peripheral and external interfaces are AMBA based, they present straightforward and well understood interfaces. This paper instead focuses on the processing cores and their inter-communication and integration. Before addressing integration, it is useful to summarize the primary processing subsystems implemented in BAZIL.

ZSP Core Subsystem:

The DSP Subsystem consists of the ZSP400 core (ZSP) and its local memory subsystem as shown in Figure 2. The ZSP400 is a 4-way superscalar, 16-bit DSP core developed by LSI Logic. The ZSP architecture based on a 5-stage pipeline made up of the following blocks [4].

1. Instruction and Data Units that manage the memory interface and implement pre-fetching of instruction and data for use by the pipeline control and execution units. The Instruction Unit (IU) does instruction pre-fetching and dispatching via a direct-mapped instruction cache in order to present four instructions per cycle to the pipeline control unit. The Data Unit (DU) does data pre-fetching, and load/ store arbitration and buffering, via a fully associative data cache. Cashing is used in the IU and DU to keep the Execution Units fed with data to maximize the number of instructions executed per cycle.

2. A Pipeline Controller Unit (PCU) that groups instructions and resolves data and resource dependencies for parallel execution. The PCU schedules instructions for execution by four functional units (2 MACs, 2 ALUs) and synchronizes pipeline operations, including operand bypass and interrupt requests.

3. The ZSP core contains two MAC and two ALU execution units. The MACs and ALUs can work independently and concurrently to perform (up to four) 16-bit by 16-bit operations per cycle. The MAC or ALU resources can be grouped for 32-bit by 32-bit operations or dual 16 bit operations.

The ZSP400 Core implements two interface ports for memory and peripherals – an Internal Port interface for close coupled, single cycle program and data memory; and an External Port for IU and DU alternative access to external memory and peripherals. The internal and external ports both contain instruction and data interfaces that support either single ported or dual ported memories

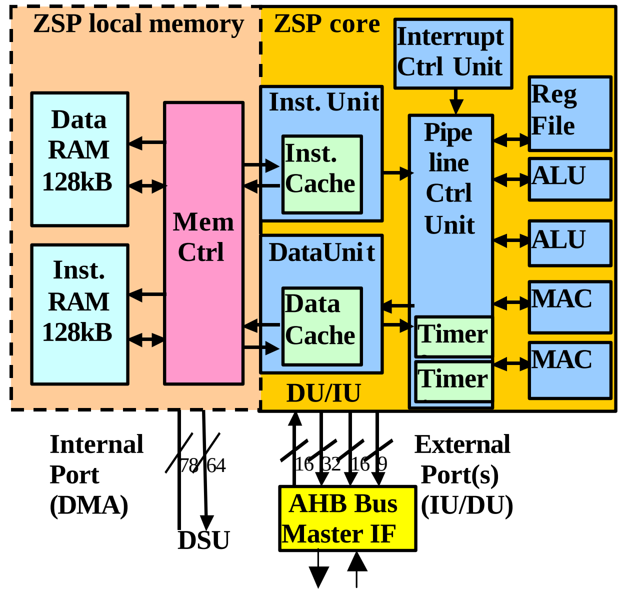

Figure 2. ZSP Core and Memory Subsystem

The Internal Port allows closely coupled “local” memory interfacing and is intended for use with synchronous on-chip memory. The ZSP core can simultaneously access internal program and data memory every cycle in order to provide data and instruction in superscalar operations. Each of the data and program memory ports support 64 bit memory reads and 32 bit writes. The internal Port I/O is non-stallable to facilitate ZSP memory throughput. By using dual ported memory and a memory interface controller (as seen in Figure 2) that allows multiplexing and segmentation of memory ports, a low overhead Direct Memory Access (DMA) interface to external on-chip logic is implemented. These DMA interfaces allow shared access by the ZSP and other logic to local ZSP subsystem memory and provide for direct high bandwidth (up to 64 bit) access of external data into the ZSP core or conversely direct export of ZSP data to external on-chip logic.

The External Port interfaces the ZSP to external memory and peripherals and provides 16 bit input and 32 bit output data bussing to the core IU and DU. The External Port interface, unlike the Internal Port interface is fully stallable. The External Port is interfaced to the AMBA AHB as a bus master, allowing control of all other blocks.

ePLC Core Subsystem:

The ePLC sub-systems are intended as loosely coupled co-processors for algorithm acceleration
The ePLC architecture is an RTL programmable logic core resource [2] developed specifically for embedded applications. The ePLC architecture is developed by Adaptive Silicon Inc. (ASi) and has been licensed by LSI Logic. The ePLC contains user configurable logic processing resource in BAZIL. Two ePLC blocks are instanced in BAZIL, both to provide flexible configuration of programmable resources (for both pre and post processing relative to DSP) and to allow for reconfigurable operations, such as one ePLC core being reprogrammed while the other is operating on data. The ePLC architecture, shown in figure 3, is made up of three parts:

1. The Multi Scale Array (MSA) contains user programmable portions of the ePLC and consists of an array of configurable ALU (CALU) Cells and their local and hierarchical interconnect and routing resources. The MSA is implemented as a hard-macro.

2. The Application Circuit Interface (ACI) provides the signal interface between the MSA and the application circuitry and is contained in the same hard-macro as the MSA. BAZIL ACIs are used for both DSU and Data buffer interfaces.

3. The PLC Adapter initiates and loads the ePLC configuration data and interfaces to test circuitry, clock and reset control through a Configuration Test Interface. BAZIL PLC Adapters integrate to an AMBA AHB slave interface. This allows the ePLC programming to he handled over the on chip AHB from flash or other external memory.

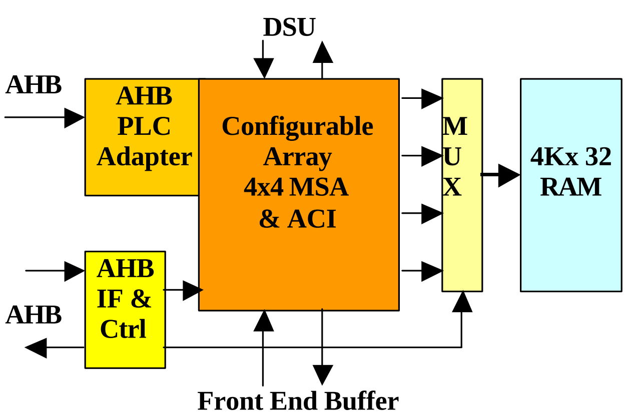

Figure 3. ePLC Core and Memory Subsystem

The ePLC contains two AHB interfaces. One, integrated with the PLC adapter, is dedicated to ePLC programming. The other AHB interface provides for general purpose communication over the AHB to peripherals and ZSP core as needed.

Figure 4 shows the internal architecture of the MSA. The MSA is made up of Hex blocks. A Hex block is the smallest physical instance used to build larger arrays. The MSA in BAZIL contains 16 Hex blocks arranged in a 4x4 configuration. A Hex in turn contains 16 Quad blocks. A Quad contains 4 CALUs. Each CALU contains 4 configurable bit-slice circuits called Function Cells and an ALU controller each of which perform a Boolean function (with optional flip-flop feedback and output). Vertically adjacent CALUs can be combined to form larger CALUs, integrating high-speed Carry-Look-Ahead and Logic Joining circuitry for complex logic and control applications.

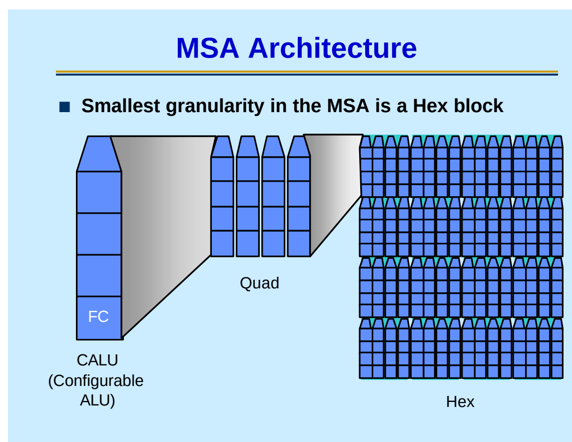

Figure 4 : Hierarchical Architecture of MSA

Hierarchical routing schemes are used for routing of the MSA both at the local Quad level and at the global MSA level. Each MSA has multiplexed access to a scratch 4Kx32 memory block.

Inter-Core Communication:

Supporting sufficient on chip bandwidth is a critical parameter in DSP/programmable logic architectures [5]. BAZIL's architecture uses dual approaches for integration between cores. Both ZSP and ePLC cores interfaces to the AMBA AHB bus, along with every other significant on-chip logic block. The AHB bus structure contains two AHB bus segments (main and external) divided by a bi-directional AHB-AHB bridge. The bus is divided by the bridge to separate high bandwidth on the external segment from low latency control traffic on the main segment.. Bridging these two types of traffic ensures they will not interfere with each other The main segment contains 3 AHB masters (ZSP, DMA and Ethernet) plus the bridge which can act as master for inter-segment communications. Control and maintenance of logic, including ePLC sub-systems is done through the main AHB.

All peripheral communication is handled through the AHB buses, with the external AHB dedicated for high bandwidth interface to system front-end, i.e. PCI, data transfers to a Front End Buffer that directly interfaces to the ePLC blocks.

AMBA does not however support levels of processor and accelerator integration desired in broadband processing. To address this, BAZIL uses a dedicated DMA/Sharing unit (DSU) interface (Figure 5) for multi-word access of ZSP internal memory data by both the ZSP and ePLC blocks. It also provides for direct data transfer between ZSP internal ports and ePLCs. This method separates high bandwidth data transfers and low latency control communication. The DSU incorporates a scheduler that shares the DMA port between ePLC accelerator sub-systems and AHB slave interface, and also handles stalling of data from the ePLC blocks when the ZSP and ePLC subsystem actively access the same memory bank in ZSP internal memory. Stalls won't occur when separate memory banks are accessed which is the preferred method.

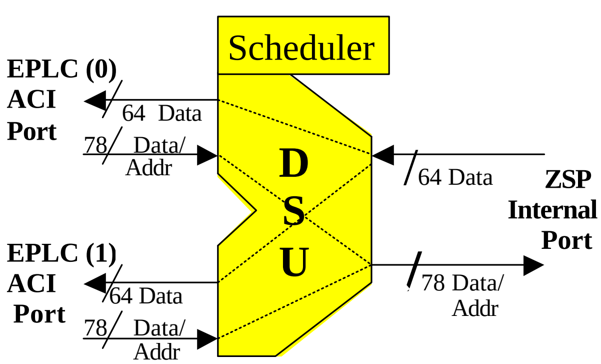

Figure 5 : DSU DMA Sharing Unit

In typical broadband processing, (Fig.6) data is imported and exported in a batch or streaming mode from a high-throughput buffered interface (in this PCI port). The data buffer simplifies the caching of bursting data on chip. ePLC blocks are used to implement a range of pre-processing and data reduction operations. Data is then presented to the ZSP, either through shared memory or directly from the DSU for DSP operation. The DSP output data can then be either exported off chip or to the ePLC for further post processing (one reason for incorporating 2 ePLC blocks) via the shared ZSP internal memory. While the DSU does not provide a communication channel between the ePLC sub systems, the ePLC systems can communicate via the shared ZSP internal memory or FEB It is also possible to move data between ePLC systems via ZSP controlled AHB traffic.

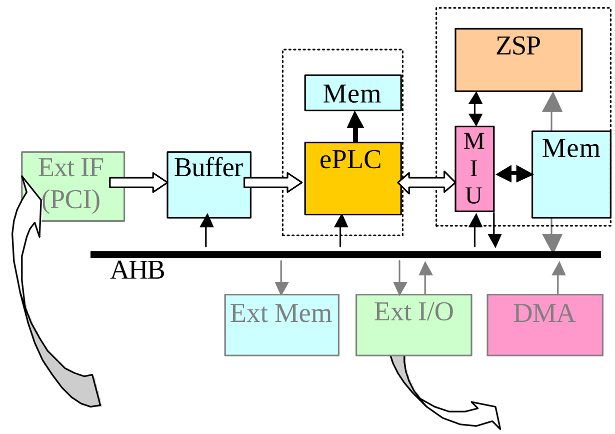

Figure 6 : BAZIL Broadband Dataflow

The amount of data available and used in different processing steps (pre-DSP and post-processing) typically is reduced with each step. As a result, interfaces required for export of processed data (Ethernet) can have significantly lower bandwidth than those needed during import stages (i.e. PCI).

BAZIL Metrics:

As seen in floorplan shown in Figure 7, largest areas of the silicon are devoted to the ePLC blocks and on-chip memory. Memory blocks (primarily ZSP and ePLC memories and Buffer FIFO) are shown in blue and soft macros, including all interface controllers, are shown in yellow.

The BAZIL architecture is clocked at 160 MHz, with the ZSP subsystems running at full clock speed. The ePLC blocks operate at 1/4 system clock speed (40 MHz). The ePLC blocks provides over 60K gates of user programmable logic, with reconfiguration controlled over the AMBA AHB, which is clocked at 80 MHz.

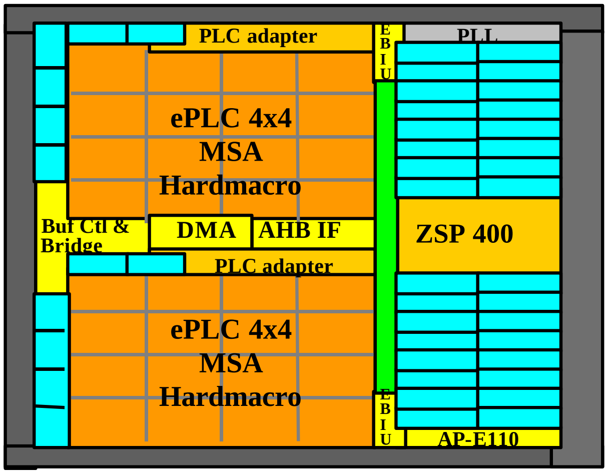

Figure 7 : BAZIL Floorplan

Summary
Heterogeneous processing platforms incorporating DSP and programmable logic cores provide an optimal mixture of resources for addressing a range of processing applications [1,5]. These cores, integrated by a novel inter-communication approach including high bandwidth interfaces (as discussed) support robust and flexible inter-core communication. The combination allows impressive levels of system programmable processor performance and reconfigurability. BAZIL, by incorporating DSP and programmable logic subsystem resources, demonstrates a new class of high processing throughput platform architectures for broadband and other high bandwidth communications applications.

Acknowledgements: The key contributions and support of the following individuals to BAZIL development are greatly appreciated: Thomas McCaughey, Brendon Slade, Pete Gasperini, and all of the Adaptive Silicon and LSI Logic ePLC design crew.

[1] "A New Direction for Computer Architecture Research" C. Kozyrakis, D. Patterson, IEEE Computer, Nov. 1998 p24-31

[2] "The Programmable Logic Core: Enabling the Configurable System-On-a-Chip", J. Hesketh, Proceedings of DesignCon2001/ PLD Forum, Feb. 2001.

[3] "Using AMBA for Signal Processor Core Integration" N. Stollon, Portable Design Magazine, July 2001

[4] "ZSP Architecture Overview White Paper" www.zsp.com

[5] "Reconfigurable Datapath and Programmable Logic Based System-on-a-Chip Architecture", N. Stollon, R. Veal, Proceedings of DesignCon2000 /IP World Forum February 2000

Neal Stollon is the Product Manager for LSI Logic's ZSP DSP development group. He has previously worked at Infinite Technology Corp., DSC/Alcatel, and Texas Instruments. He has a Ph.D in EE from Southern Methodist University and is a registered Professional Engineer (Tx.).

Björn Sihlbom is Systems Engineer for Ericsson Microwave Systems ASIC Technology & System on Silicon group. He holds a M. Sc. E.E. from Chalmers University of Technology in Sweden.