5

Current FPGA Technologies

5.1 Introduction

The analysis at the end of previous chapter makes it clear that the choice of the specific technology and the resulting design approach directly impacts the performance that will be achieved. For example, the use of simple DSP microcontrollers typically implies a DSP system with relatively low performance requirements such as medical or industrial control systems. The design effort is only that needed to produce efficient C or C++ source code for its implementation, and indeed it may be possible to use the software compilers associated with MATLAB® or LabVIEW that the user may have used as the initial design environment to scope the requirements such as wordlength and system complexity.

This design approach can be applied for the full range of "processor"- style platforms, but it may be required that dedicated handcrafted C code is produced to achieve the necessary performance. This is probably particularly relevant in applications where performance requirements are tight. Also, the hardware may possess dedicated functionality that is not well supported within the high-level tool environment. In these cases, it is clear that the platform will be chosen to meet some superior area, speed and power performance criteria.

Mindspeed's T33xx family of wireless application processors (Mindspeed 2012) have been directly targeted at base stations for mobile services and thus have dedicated functionality such as forward error correction (FEC) and Mindspeed application processor DSP blocks for advanced signal processing and encryption functions. In these cases, the user has to compromise on ease of design, in order to take advantage of the specific architectural feature offered by the technology or use company-specific tool sets to achieve the necessary performance. This notion is taken to the extreme in the SoC arena where the user is faced with creating the circuit architecture to best match the performance requirements.

Working towards this ultimate performance is effectively what an FPGA platform offers. From "glue logic" beginnings, FPGAs have now become an advanced platform for creating high- performance, state-of-the-art systems. Indeed, many high-performance data-processing environments are starting to see the benefits of this technology. The

FPGA-based Implementation of Signal Processing Systems,
Second Edition. Roger Woods, John McAllister, Gaye Lightbody and Ying Yi.

purpose of this chapter is to give a review of the current FPGA technologies with a focus on how they can be used in creating DSP systems. The chapter acts to stress key features and leaves the detail to the vendors' data sheets. Whilst a number of technologies are available, the focus is on the latest commercial offerings from the two dominant vendors, namely Xilinx and Altera, whilst giving a brief description of the other solutions.

Section 5.2 gives a brief historical perspective on FPGAs, describing how they have emerged from being a fine-grained technology to a complex SoC technology. Section 5.3 describes the Altera Stratix® 10 FPGA family, the most powerful FPGA family that the company offers. The next sections then go on to describe the FPGA technology offerings from Xilinx, specifically the UltraScale™ (Section 5.4) and Zynq® (Section 5.5) FPGA families. The technologies offered by Microsemi and Lattice offer specific features that are very relevant in certain markets. For this reason, Lattice's iCE40isp family of small, low-power, integrated mobile FPGAs is described in Section 5.6, and Microsemi's RTG4, a radiation tolerant (RT) FPGA for signal processing applications, is described in Section 5.7. Section 5.8 attempts to summarize the key features of recent FPGA devices and gives some insights into FPGA-based DSP system design. Some conclusions are given in Section 5.9.

5.2 Toward FPGAs

In the 1970s, logic systems were created by building PCB boards consisting of transistor-transistor logic (TTL) logic chips. However, as functions got larger, the logic size and levels increased and thus compromised the speed of design. Typically, designers used logic minimization techniques, such as those based on Karnaugh maps or Quine–McCluskey minimization, to create a sum of products expression by generating the product terms using AND gates and summing them using an OR gate.

The concept of creating a structure to achieve implementation of this functionality was captured in the early programmable array logic (PAL) device, introduced by Monolithic Memories in 1978. The PAL comprised a programmable AND matrix connected to a fixed OR matrix which allowed sum of products structures to be implemented directly from the minimized expression. The concept of an AND and OR matrix became the key feature of a class of devices known as programmable logic devices (PLDs); a brief classification is given in Table 5.1, the final member of which is of course the ROM.

As illustrated in Figure 5.1, a ROM possesses the same structure only with a fixed AND plane (effectively a decode) and a programmable OR plane. In one sense, an $n \times m$ structure can be viewed as providing the capability of storing four (in general, n^2) two-bit (or m-bit) words, as shown in Figure 5.2. The decoder, which is only required to reduce the number of pins coming into the memory, is used to decode the address input pins,

Table 5.1 PLD types

	AND matrix	OR matrix
ROM	Fixed	Programmable
PLA	Programmable	Fixed
PAL	Programmable	Programmable

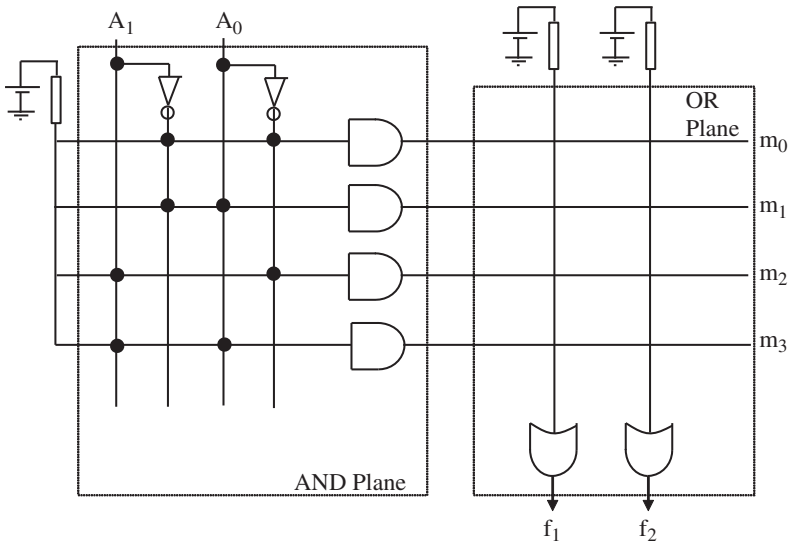

Figure 5.1 PLD architecture

and a storage area or memory array is used to store the data. As the decoder generates the various address lines using AND gates and the outputs are summed using OR gates, this provides the AND–OR configuration needed for Boolean implementation.

In general, as an $n \times 1$-bit ROM could implement any n-input Boolean function, a four-input ROM or LUT became the core component of the very first FPGA, the Xilinx XC2000. The four-input LUT was small enough to achieve efficient utilization of the chip area, but large enough to implement a reasonable range of functions, based on early analysis (Rose *et al.* 1990). If a greater number of inputs is required, then LUTs are cascaded together to provide the implementation, but at a slower speed. This was judged to provide an acceptable trade-off.

The PLD structure had a number of advantages. It clearly matched the process of how the sum of products was created by the logic minimization techniques. The function could then be fitted into one PLD device, or, if not enough product terms were available, then it could be fed back into a second PLD stage. Another major advantage was that the circuit delay is deterministic, either comprising one logic level or two, etc. However,

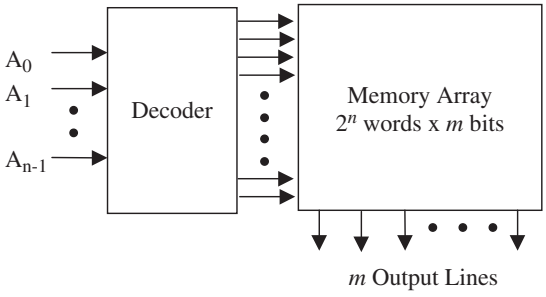

Figure 5.2 Storage view of PLD architecture

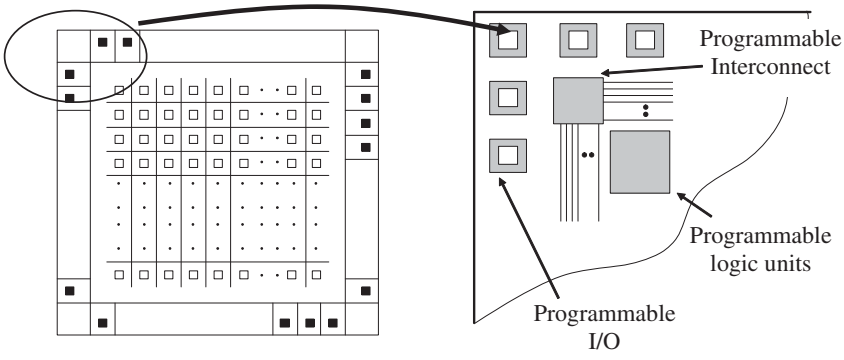

Figure 5.3 Early Manhattan architecture

the real advantage comes in the form of the programmability which reduces the risk in PCB development, allowing possible errors to be fixed by adjusting the PLD logic implementation. But as integration levels grew, the idea of using the PLD as a building block became an attractive FPGA proposition as illustrated by the early Altera MAX® family. As mentioned earlier, Xilinx opted for the LUT approach.

5.2.1 Early FPGA Architectures

The early FPGA offerings were based around the Manhattan-style architecture shown in Figure 5.3 where each individual cell comprised simple logic structures and cells were linked by programmable connections. Thus, the FPGA could be viewed as comprising the following:

- programmable logic units that can be programmed to realize different digital functions;
- programmable interconnect to allow different blocks to be connected together;
- programmable I/O pins.

This was ideal for situations where FPGAs were viewed as glue logic as programmability was then the key to providing redundancy and protection against PCB board manufacture errors; it might even provide a mechanism to correct design faults. However, technology evolution, outlined by Moore's law, now provided scalability for FPGA vendors. During the 1980s, this was exploited by FPGA vendors in scaling their technology in terms of numbers of levels of interconnectivity and number of I/Os. However, it was recognized that this approach had limited scope, as scaling meant that interconnect was becoming a major issue and technology evolution now raised the interesting possibility that dedicated hardware could be included, such as dedicated multipliers and, more recently, ARM™ processors. In addition, the system interconnectivity issue would be alleviated by including dedicated interconnectivity in the form of Serializer/Deserializer (SERDES) and RapidIO. Technology evolution has had a number of implications for FPGA technology:

- **Technology debate** In the early days, three different technologies emerged, namely conventional SRAM, anti-fuse and electrically erasable programmable read-only memory (E^2PROM) technologies. The latter two technologies both require special

steps to create either the anti-fuse links or the special transistors to provide the E^2PROM transistor. Technological advances favored SRAM technology as it required only standard technology: this became particularly important for Altera and Xilinx, as FPGA fabrication was being outsourced and meant that no specialist technology interaction with the silicon fabrication companies was needed. Indeed, it is worth noticing that silicon manufacturers now see FPGA technologies as the most advanced technology to test their fabrication facilities.

- **Programmable resource functionality** A number of different offerings again exist in terms of the basic logic block building resource used to construct systems. Early offerings (e.g. Algotronix and Crosspoint) offered simple logic functions or multiplexers as the logic resource, but with interconnect playing an increasing role in determining system performance, these devices were doomed. Coarser-grained technologies such as the PLD-type structure (and, more particularly, the LUT), dominated because they are flexible and well understood by computer programmers and engineers. Examining the current FPGA offerings, it is clear that the LUT-based structure now dominates with the only recent evolution an increase in the size of the LUT from a four-input to a five- or six-input version in the Xilinx Virtex/ UltraScale™ technology and to an eight-input version in the Altera Stratix® family.
- **Change in the FPGA market** Growing complexity meant that the FPGA developed from being primarily a glue logic component to being a major component in a complex system. With DSP being a target market, FPGA vendors had to compare their technology offerings in terms of new competitors, primarily the DSP processor developers such as TI and Analog Devices presented in Chapter 4.
- **Tool flow** Initially, FPGAs were not that complex, so up until the mid 1990s, tools were basic. Eventually they had to become more complex, moving toward automatic place and route tools. These still play a major role in vendors' tool flows. With increasing complexity, there has been an identified need for system-level design tools: DSP Builder and SDK for OpenCL from Altera, and Vivado and the SDSoC™ development environment from Xilinx. This may be an increasingly problematic issue as tools tend to lag well behind technology developments; it is a major area of focus in this book.

It has now got to the stage that FPGAs represent system platforms. This is recognized by both major vendors who now describe their technology in these terms. Xilinx describes its Zynq® UltraScale+™ FPGA as a technology comprising heterogeneous multi-processing engines, while Altera describes its Stratix® 10 FPGA family as featuring its third-generation hard processor system. Thus, the technology has moved from the era of programmable cells connected by programmable interconnect, as highlighted at the start of this section, to devices that are complex, programmable SoCs which comprise a number of key components, namely dedicated DSP processor blocks and processor engines.

5.3 Altera Stratix® V and 10 FPGA Family

Altera offers a series of FPGAs covering a range of performance needs and application domains (see Table 5.2). Its leading-edge FPGA is the Stratix® 10. Details were

Table 5.2 Altera FPGA family

Family	Brief description
Stratix®	High-performance FPGA and SoC family with multiple DSP blocks embedded memory, memory interfaces, transceiver blocks which supports partial reconfiguration
Arria	Focused on power efficiency, has a "hard" floating-point DSP block, support for 28.3 Gbps and "smart voltage" capability
MAX 10	A non-volatile technology that has programmable logic, DSP blocks and soft DDR3 memory controller and supports dynamic reconfiguration

not widely available at the time of writing, so most of the following discussion is based around the architecture of Stratix® V as many of the architectural features would appear to have remained the same.

The FPGA architecture has evolved from the early Manhattan-style tiling of LUTs/flip-flop cells (see Figure 5.3), into columns of programmable logic, dedicated DSP silicon blocks and scalable memory blocks. Also included are dedicated PLLs, embedded peripheral component interconnect (PCI) express bus standard, transceivers and general-purpose I/Os. The core components include adaptive logic modules (ALMs), DSP blocks and memory, covered below.

5.3.1 ALMs

The ALM is the core programmable unit and extends the basic concept of the four-input LUT and D-type flip-flop which has been the core FPGA programmable part for many years. As shown in Figure 5.4, it contains LUT-based resources that can be divided between two combinational adaptive lookup tables (ALUTs) and four registers, allowing various configurations.

With up to eight inputs for the two combinational ALUTs, one ALM can implement various combinations of two functions, allowing backward compatibility with the older four-input LUT architectures. One ALM can also implement any function with up to six-input and certain seven-input functions. After many years of four-input LUTs, it has now been deemed viable to use large LUT sizes. In addition, the user can configure the ALM as a simple dual-port SRAM in the form of a 64×1 or a 32×2 block. The ALM output can be registered and unregistered versions of the LUT or adder output. The register output can also be fed back into the LUT to reduce fan-out delay.

It also contains four programmable registers, each with the functionality of clocks, synchronous and asynchronous clear, synchronous load and circuitry to drive signals to the clock and clear control signals; this providing a very wide range of functionality. The registers can be bypassed and the output of the LUT can directly drive the ALM outputs.

These ALMs are contained within logic array blocks (LABs), and the LAB contains dedicated logic for driving control signals to its ALMs; it has two unique clock sources and three clock enable signals. LAB-wide signals control the register logic using two clear signals, and the Stratix® V device has a device-wide reset pin that resets all the registers in the device.

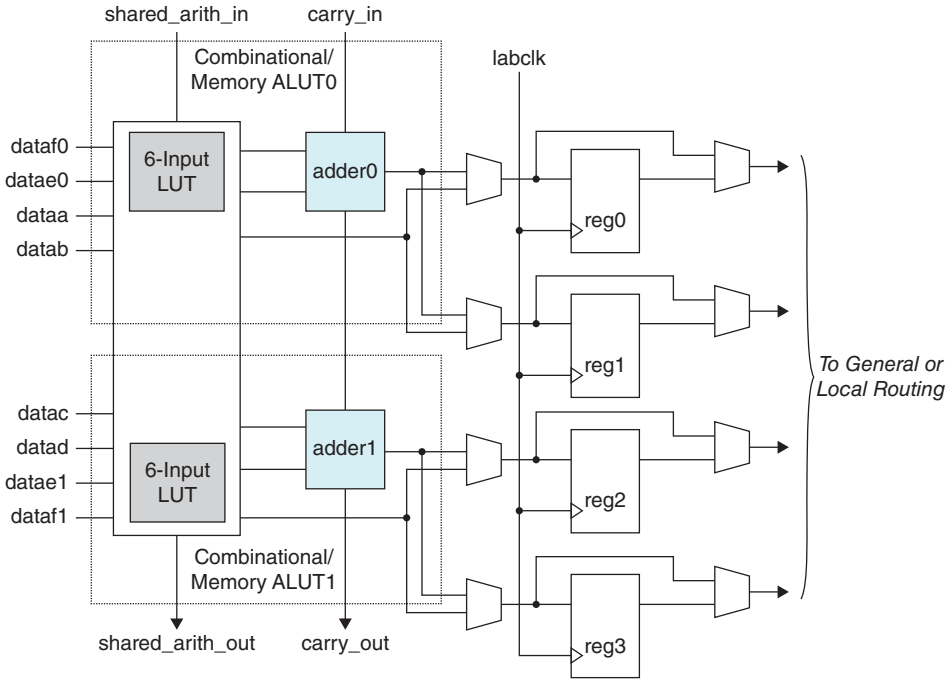

Figure 5.4 Adaptive logic module

5.3.2 Memory Organization

The Stratix® V device contains two types of memory blocks each of which can be clocked at 600 MHz:

- *M20K* The Stratix® V contains up to 11.2 GB of 20 kB blocks of dedicated memory resources. The M20K blocks are ideal for larger memory arrays while still providing a large number of independent ports.
- *640-bit memory LABs (MLABs)* These are memory blocks that are configured from the LABs and can be configured as wide or shallow memory arrays for use as shift registers, wide shallow first-in, first-out (FIFO) buffers, and filter delay lines. For the Stratix® V devices, the ALMs can be configured as ten 32×2 blocks, giving a 32×20-bit or 64×10-bit simple dual-port SRAM block per MLAB.

The mixed-width port configuration is supported in the simple and true dual-port RAM memory modes in various configurations (Altera 2015a). The aim is to provide many modes of operation of the memory hierarchy to support as much functionality as possible, a summary of which is given in Table 5.4. With the streaming nature of DSP and image processing systems, there is a need to delay data, which is typically achieved using shift registers and FIFOs, both of which are supported, and for many DSP operations there is a need to store fixed coefficient values, which the ROM and indeed RAM mode of operation will permit. In more complex modes of operation, there is a need to access and change temporary information, which is supported in the various RAM

Table 5.3 Supported embedded memory block configurations

Memory	Block depth (bits)	Programmable width
MLAB	32	× 16, × 18, or × 20
	64	× 8, × 9, × 10
M20K	512	× 40, × 32
	1K	× 20, × 16
	2K	× 10, × 8
	4K	× 5, × 4
	8K	× 2
	16K	× 1

mode configurations. It is the FPGA vendor's goal to provide underlying support for the many modes of DSP functionality required.

Various clock modes are supported for the memory. This includes *single*, where a single clock controls all registers of the memory block; *read/write*, where a separate clock is available for each read and write port; *input/output*, where a separate clock is available for each input and output port; and *independent*, where a separate clock is available for each port (A and B).

Bit parity checking is supported where the parity bit is the fifth bit associated with every four data bits in data widths of 5, 10, 20, and 40. The error correction code (ECC) support provided allows detection and correction of data errors at the output of the memory, providing single, double-adjacent and triple-adjacent error detection in a 32-bit word.

5.3.3 DSP Processing Blocks

Each variable-precision DSP block spans one LAB row height and offers a range of multiplicative and additive support functionality targeted at support for DSP functionality, specifically 9-bit, 18-bit, 27-bit, and 36-bit wordlength support and even based around an 18 × 25 multiplier block with built-in addition, subtraction, and 64-bit accumulation unit to combine multiplication results.

Table 5.4 Memory configurations

Memory mode	Brief description
Single-port RAM	Supports one read/one write operation. Read enable port can be used to show the previous held values during the most recent active read enable or to show new data being written
Simple dual-port RAM	One read on port B and one write on port A, i.e. different locations
True dual-port RAM	Any combination of two port operations: two reads, two writes, or one read and one write at two different clock frequencies
Shift register	Can create $w \times m \times n$ shift register for input data width (w), tap size (m), and tap number (n). Can also cascade memory blocks
ROM	Can use memory as ROM. Various configuration of registered and unregistered address lines and outputs
FIFO buffers	Allow single and dual clock asynchronous FIFO buffers

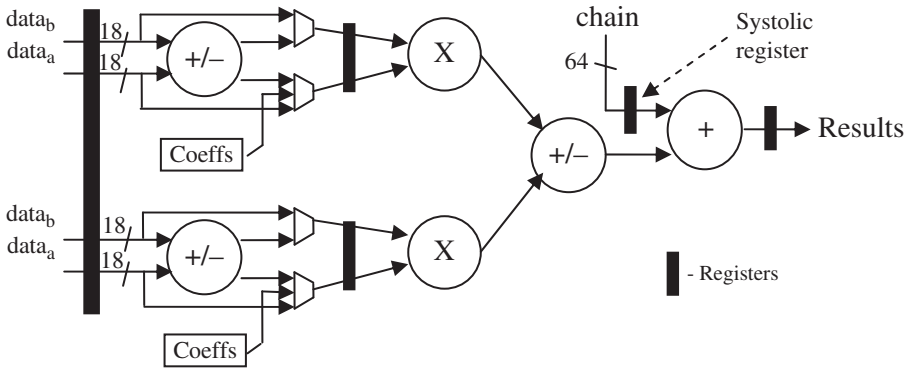

Figure 5.5 Altera DSP block (simplified view)

As illustrated in Figure 5.5, the Stratix® V variable- precision DSP block consists of the following elements:

- input register bank to store the various input data and dynamic control signals;
- pre-adder which supports both addition and subtraction;
- internal coefficient storage which can support up to eight constant coefficients for the multiplicands in 18-bit and 27-bit modes;
- multipliers which can be configured as a single 27 × 27-bit multiplier, two 18 × 18-bit multipliers or three 9 × 9-bit multipliers;
- accumulator and chainout adder which supports a 64-bit accumulator and a 64-bit adder;
- systolic registers, to register the upper multiplier's two 18-bit inputs and to delay the chainout output to the next variable- precision DSP block;
- 64-bit bypassable output register bank.

The processing architecture is highly programmable, allowing various functions to be cascaded together or, in some cases, bypassed. This includes the input and output registers and those referred to as systolic registers which can be used for pipelining, acting to decrease the critical path but increase the latency. This will be achieved during the place process in the synthesis flow using the Altera Quartus® design software. The architecture of the block has been created in such a way as to make it cascadable, thereby allowing core DSP functions to be implemented, such as a FIR filter which comprises a classical multiplier and adder tree function.

Each DSP block can implement one 27 × 27-bit multiplier, two 18 × 18-bit multipliers or three 9 × 9-bit multipliers. It can also be made to perform floating-point arithmetic through the use of additional functionality. The multiplier can support signed and unsigned multiplication and can dynamically switch between the two without any loss of precision. The DSP block can also be configured to operate as a complex multiplier.

The unit can be configured as an adder, a subtracter, or as an accumulator, based on its required mode of operation, and has been designed to automatically switch between adder and subtracter functionality. The DSP blocks have been co-located with the dedicated embedded memory devices to allow coefficients and data to be effectively stored and for memory-intensive DSP applications to be implemented.

5.3.4 Clocks and Interconnect

The Stratix® architecture is organized with three clock networks that are fixed in a hierarchical structure of global clocks which are configured in an H tree structure to balance delay; regional clocks which give low clock skew for logic contained within that quadrant; and periphery clocks which have higher skew than the other clocks. The clock utilizes the Altera MultiTrack interconnect which provides low-skew clock and control signal distribution. It consists of continuous, performance-optimized routing lines of different lengths and speeds used for inter- and intra-design block connectivity.

The FPGA also provides robust clock management and synthesis for device clock management, external system clock management, and high-speed I/O interfaces. It also contains up to 32 fractional phase locked loops (PLLs) that can function as fractional PLLs or integer PLLs, and output counters are dedicated to each fractional PLL that support integer or fractional frequency synthesis. This system of clocking and PLLs should be sufficient to provide low-latency synchronous connections required in many DSP systems.

5.3.5 Stratix® 10 innovations

Stratix® 10 builds on the Stratix® V architecture but with improved DSP performance. It is manufactured on the Intel 14 nm tri-gate process which improves planar transistor technology by creating a "wraparound" gate on the source-to-drain "channel," and gives better performance, reduces active and leakage power, gives better transistor density and a reduction in transistor susceptibility to charged particle single event upsets (SEUs). It offers improved floating-point arithmetic performance and improved bandwidth via the new HyperFlex™ architecture (Altera 2015b). The technology also comprises a 64-bit quad-core ARM™ CortexTM-A53, integrated 28.05- and 14.1-Gbps transceivers, up to 6 × 72 DDR3 memory interfaces at 933 MHz and 2.5 TMACS of signal processing performance.

HyperFlex™ is a high-speed interconnection architecture that allows users to employ "hyper-registers" which are associated with each individual routing segment in the device and can allow the speed to be improved; they can also be bypassed. The registers are also available at the inputs of all functional blocks such as ALMs, M20K blocks, and DSP blocks. The concept is based around pipelining to avoid long interconnect delays by employing retiming procedures without the need for user effort. It results in an average performance gain of 1.4× for Stratix® 10 devices compared to previous generation high-performance FPGAs.

The Stratix® 10 device offers an improved DSP block which in larger devices can deliver up to 11.5 TMAC or 23 TMAC when using the pre-adder and 9.3 TFLOPS of single-precision, floating-point performance using dedicated hardened circuitry (Altera 2015b). The fixed-point functionality operates at 1 GHz and its floating-point modes operate at 800 MHz. The company argues that this gives a superior power efficiency of 80 GFLOPS/W when compared to GPUs.

5.4 Xilinx Ultrascale™/Virtex-7 FPGA families

Xilinx's latest FPGA technology is centered on its UltraScale™ family which has been implemented in both a 16 nm and a 20 nm CMOS technology using planar

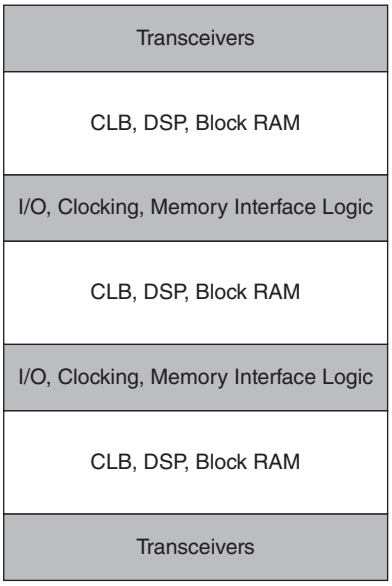

Figure 5.6 Xilinx Ultrascale™ floorplan. Reproduced with permission of Xilinx, Incorp.

and FinFET technologies and 3D scaling. In addition to the classical FPGA offerings, there is a strong focus on multi-processing SoC (MPSoC) technologies. In addition, multiple integrated ASIC-class blocks for 100G Ethernet, 150G Interlaken, and PCIe Gen4 are available to allow fast and efficient data transfer into the FPGA. Static and dynamic power gating is available to address the increasing energy concerns. Advanced encryption standard (AES) bitstream decryption and authentication, key obfuscation, and secure device programming are also included to address security aspects.

The family comprises the Kintex® UltraScale+™ which is focused on low power and the Virtex® UltraScale+™ which is focused on performance. The family also includes the Zynq® UltraScale+™ which incorporates hardware processing technologies in the form of ARM™ processors. UltraRAM has been incorporated to provide larger on-chip SRAM memory. DDR4 can support up to 2666 Mb/s for massive memory interface bandwidth.

The FPGA resources are organized in columns as indicated in Figure 5.6. These include CPUs, DSP blocks called DSP48E2 components and block random access memory (BRAM). High-speed transceivers are used to get the data in and out of the FPGA device quickly, in addition to clocking and memory interfacing circuitry. Xilinx has utilized a new form of routing and an ASIC-like clocking to improve performance. The DSP block has been tweaked to allow better fixed-point and IEEE Standard 754 floating-point arithmetic.

5.4.1 Configurable Logic Block

The Xilinx configurable logic block (CLB) comprises a number of six-input functional generators, flip-flops, fast-carry logic for adder implementation and various programmable hardware to allow various configurations to be created. One quarter of the slice is shown in Figure 5.7 and gives an idea of the core functionality. The six-input

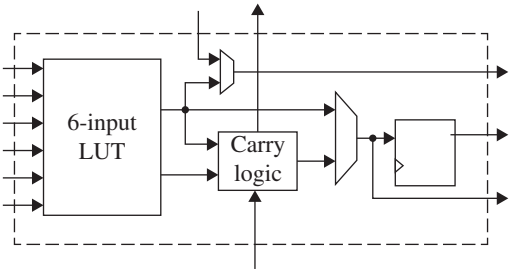

Figure 5.7 Simplified view of 1/4 Xilinx slice functionality

function generator can realize a six-input Boolean function or two arbitrarily defined five-input Boolean functions, as long as these two functions share common inputs. The propagation delay through a LUT is independent of the function implemented, and signals from function generators can connect to slice outputs or to other resources within the slice.

As with the Altera ALM, the six-input functional generator can be configured as a synchronous or distributed RAM. Multiple LUTs in a slice can be combined in various ways to store larger amounts of data up to 256 bits per CLB. Multiple slices can be combined to create larger memories, either single-port RAM (32×1–16-bit, 64×1–8-bit, 128×1–4-bit, 256×1–2-bit, 512×1-bit), a dual-port RAM (32×1–4-bit, 64×1–4-bit, 128×2-bit, 256×1-bit), a quad-port RAM (32×1–8-bit, 64×1–2-bit, 128×1-bit), an octal-port (64×1-bit) and a simple dual-port RAM (32×1–14-bit, 64×1–7-bit).

The availability of the registers means that sequential circuitry can be implemented by connecting the six-input functional generator to the flip-flops and thus realizing counters or a finite state machine. This allows for a range of controllers to be created that are typically used to organize dataflow in DSP systems.

5.4.2 Memory

As with the other main FPGA vendor, Xilinx offers a range of memory sizes and configurations. The UltraScale^{TM} architecture's memory is organized into 36 kB block RAMs, each with two completely independent ports that share only the stored data (Xilinx 2015a). Like the DSP blocks, the memory is organized into columns as shown in Figure 5.6. Each block can be configured as a single 36 kB RAM or two independent 18 kB RAM blocks. Memory accesses are synchronized to the clock, and all inputs, data, address, clock enables, and write enables are registered with an option to turn off address latching. An optional output data pipeline register allows higher clock rates at the cost of an extra cycle of latency.

BRAMs can be configured vertically to create large, fast memory arrays, and FIFOs with greatly reduced power consumption. BRAM sites that remain unused in the user design are automatically powered down and there is an additional pin on the BRAM to control the dynamic power gating feature. The BRAMs can be configured as $32K \times 1$-bit, $16K \times 2$-bit, $8K \times 4$-bit, $4K \times 9$ (or 8)-bit, $2K \times 18$ (or 16)-bit, $1K \times 36$ (or 32)-bit, or 512×72 (or 64)-bit. The two ports can have different aspect ratios without any constraints. Moreover, as each block RAM can be organized as two 18 kB block RAMs, they be configured to any aspect ratio from $16K \times 1$ to 512×36-bit.

Only in simple dual-port (SDP) mode can data widths greater than 18 bits (18 kB RAM) or 36 bits (36 kB RAM) be accessed. This may have implications for how the memory will be used in the prospective DSP system. In this mode, one port is dedicated to read operation, the other to write operation. In SDP mode, one side (read or write) can be variable, while the other is fixed to 32/36 or 64/72. Both sides of the dual-port 36 kB RAM can be of variable width.

The memory also has an error detection and correction and each 64-bit-wide BRAM can generate, store, and utilize eight additional Hamming code bits and perform single-bit and double-bit error detection during the read process. The ECC logic can also be used when writing to or reading from external 64- to 72-bit-wide memories.

5.4.3 Digital Signal Processing

Each DSP slice fundamentally consists of a dedicated 27×18-bit two's complement multiplier and a 48-bit accumulator (Xilinx 2015c) as shown in Figure 5.8. The multiplier can be dynamically bypassed, and two 48-bit inputs can feed a SIMD arithmetic unit (dual 24-bit add/subtract/accumulate or quad 12-bit add/subtract/accumulate), or a logic unit that can generate any one of ten different logic functions of the two operands. The DSP slice includes an additional pre-adder which improves the performance in densely packed designs and reduces the DSP slice count by up to 50%. The 96-bit-wide XOR function, programmable to 12, 24, 48, or 96-bit wide, enables performance improvements when implementing FEC and cyclic redundancy checking algorithms. The DSP also includes a 48-bit-wide pattern detector that can be used for convergent or symmetric rounding. The pattern detector is also capable of implementing 96-bit-wide logic functions when used in conjunction with the logic unit.

The DSP block contains the following components:

- a 27×18 two's-complement multiplier with dynamic bypass;
- a power-saving 27-bit pre-adder;
- a 48-bit accumulator that can be cascaded to build 96-bit and larger accumulators, adders, and counters;
- an SIMD arithmetic unit, namely a dual 24-bit or quad 12-bit add/subtract/accumulate;

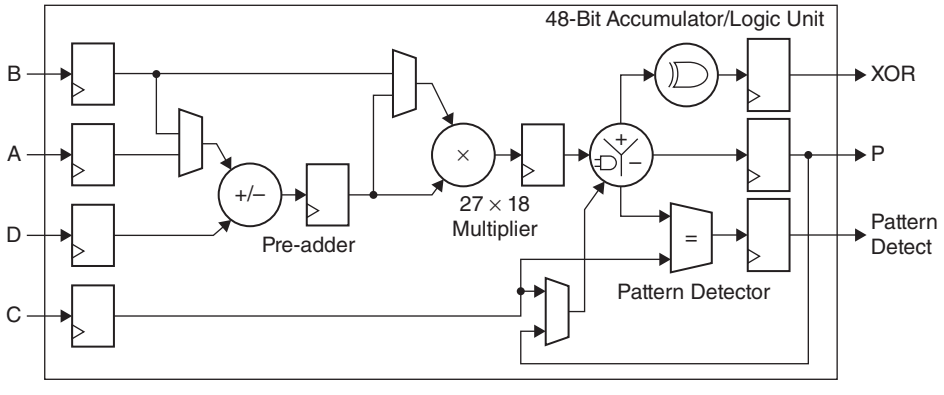

Figure 5.8 Xilinx DSP48E2 DSP block. Reproduced with permission of Xilinx, Incorp.

- a 48-bit logic unit that can implement bitwise AND, OR, NOT, NAND, NOR, XOR, and XNOR functions;
- a pattern detector that allows a number of features including terminal counts, overflow/underflow, convergent/symmetric rounding support, and wide 96-bit wide AND/NOR when combined with logic unit to be performed;
- optional pipeline registers and dedicated buses for cascading multiple DSP slices in a column for larger functions.

Whilst the DSP48E2 slice is backward compatible with older technologies, a number of minor changes have been made to the previous DSP48E1 block, including wider functionality in the slice including increased word size in the multiplier from 25 to 27 bits and accompanying size increase in the pre-adder. The number of operands to the ALU has been increased and means of cascading blocks have been improved.

As has been indicated previously, it is important to have a core understanding of the underlying technology when implementing DSP functionality. For use of the DSP48E2 block, pipelining is encouraged for both performance as it decreases the critical path at the expense of increased latency (see Chapter 9) and power as pipelining reduces the switched capacitance aspect of dynamic power consumption by both reducing the routing length and the switching activity due to less routing (see Chapter 13).

5.5 Xilinx Zynq FPGA Family

The Xilinx Zynq merits a separate section as it represents a new form of FPGA-based system that makes it highly suitable for achieving a new form of computing architecture, historically referred to as an FPGA-based custom computing machine (FCCM) or reconfigurable computing. A simplified version of the Xilinx Zynq architecture is given in Figure 5.9. It comprises a dedicated ARM^{TM} processor environment which has been called a processing system (PS) connected to the standard programmable logic (PL) which is the same as that for the Xilinx UltraScale^{TM}. As the PL has been just described, the description will concentrate on the PS aspects.

The Zynq processing system has an ARM^{TM} processor and a set of resources which form the application processing unit (APU) comprising peripheral interfaces, cache memory, memory interfaces, interconnect, and clock generation circuitry (Crockett *et al.* 2014). The system is shown in Figure 5.10 and is composed of two ARM^{TM} processing cores, each of which has a NEON^{TM} media processing engine (MPE) and

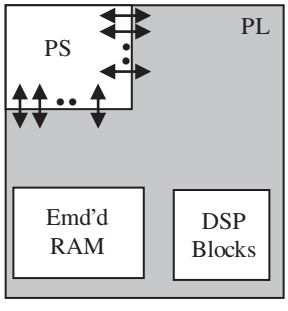

Figure 5.9 Xilinx Zynq architecture

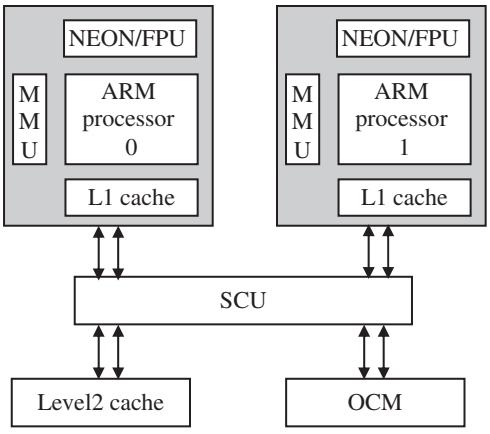

Figure 5.10 Zynq application processing unit

floating-point unit (FPU), a level 1 cache memory and a memory management unit (MMU) unit. The APU also contains a level 2 cache memory, and a further on-chip memory (OCM). Finally, a snoop control unit (SCU) forms a bridge between the ARM^{TM} processors and the level 2 cache and OCM memories and also interfaces to the PL.

The ARM^{TM} can operate at up to 1 GHz, and each has separate level 1 caches for data and instructions, both of which are 32 kB. This allows local storage of frequently required data and instructions for fast processor performance. The two cores additionally share a level 2 cache and there is a further 256 kB of on-chip memory within the APU. The MMU allows translation between virtual and physical addresses.

The Zynq technology has been applied to a number of image processing and evolvable applications. A Zyng-based FPGA-based traffic sign recognition system has been developed for driver assistance applications (Siddiqui *et al.* 2014; Yan and Oruklu 2014). Dobai and Sekanina (2013) argue that the technology has the potential to become "the next revolutionary step in evolvable hardware design."

As FPGAs become more popular in computing systems, there is no doubt that these type of architectures will become popular. Computing vendors develop memory interfaces to directly read from and write to the FPGA directly, such as IBM's coherent accelerator processor interface (CAPI). The use of FPGAs to process data will only become prevalent if data can be quickly loaded and offloaded.

5.6 Lattice iCE40isp FPGA Family

Lattice Semiconductor offers a range of low-power and low-cost products. Its technologies include those listed below:

- MachXO is a non-volatile technology that includes multi-time programmable non-volatile configuration memory and infinitely reconfigurable flash memory. It is contained in small wafer-level chip-scale packaging and is available with low-voltage cores. They comprise mainly LUTs, DRAM and SRAM and various other supporting functionality.

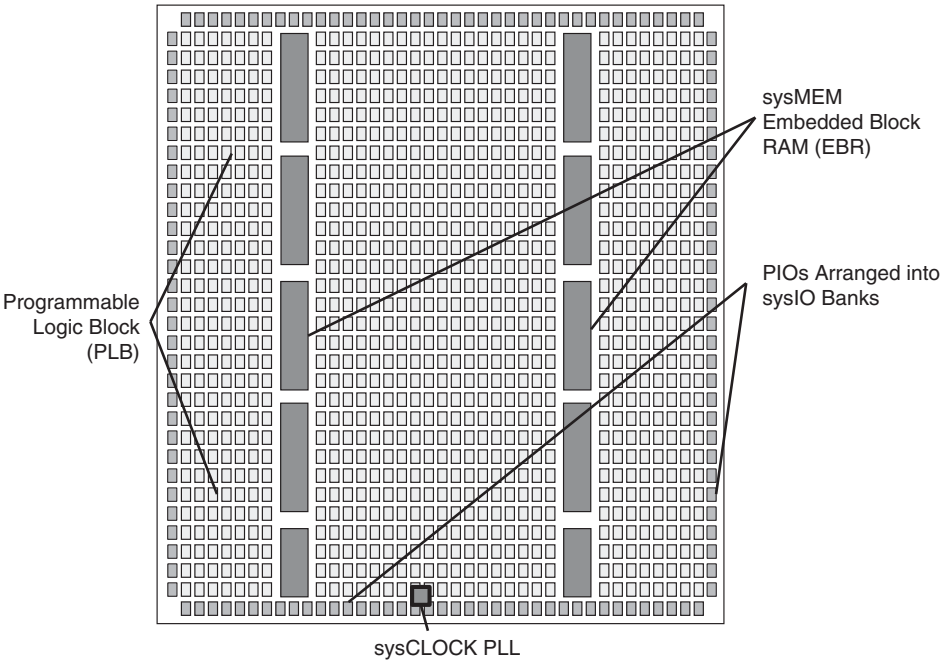

Figure 5.11 Lattice Semiconductor iCE40isp. Reproduced with permission of Lattice Semiconductor Corp.

- LatticeECP3 is mainly a communications FPGA targeted at industrial, telecommunications or automotive infrastructure equipment; it implements SERDES giving up to 16 channels at 3.125 Gbps and 800 Mbps DDR3 access.
- iCE40 is a family of ultra-low power, non-volatile FPGAs fabricated in a 40 nm CMOS low-power process and targeted at a range of hand-held and battery-powered applications.

The iCE40isp is chosen as the focus here as it is targeted at low power consumption applications and represents an alternative when compared to the earlier FPGA families. It comprises ultra-low density (ULD) devices with small power consumption. It has five devices with densities ranging from 384 to 7680 LUTs arranged into programmable logic blocks (PLBs), along with DSP functional blocks, embedded block random access memory (EBR) and PLLs. The architecture (see Figure 5.11) is comprised of a tiling of PLBs connected to a number of EBRs. There are also Inter-Integrated Circuit (I2C) and Serial Peripheral Interface (SPI) cores that enable flexible device configuration through SPI which gives advantages in interfacing for image processing. Static and dynamic power consumption reduction is achieved by programmable low-swing differential I/Os and turning off on-chip PLLs dynamically.

5.6.1 Programmable Logic Blocks

The Lattice iCE40isp is comprised of PLBs, containing eight logic cells each of which includes three primary logic elements as outlined below:

- a four-input LUT which allows any four-input combinational logic function to be implemented or cascaded of any complexity and also implemented as a 16×1 ROM;
- a D-type flip-flop with an optional clock enable and reset control input allowing configuration for a range of sequential circuits and memory storage;
- carry logic for fast arithmetic functions allowing fast arithmetic circuits.

This architecture is similar to the earlier Altera and Xilinx technologies and would therefore be expected to give high levels of performance for a wide of application domains. Given the need to use the fast adder combination for DSP implementation, the design performance will benefit from many of the techniques highlighted in Chapter 6.

5.6.2 Memory

Using programmable logic resources, an EBR implements a variety of logic functions, each with configurable input and output data widths. The EBR is the embedded block RAM of the device, each 4 kB in size, and can be configured to create a number of memory configurations and sequential digital circuits (Lattice 2015):

- random-access memory (RAM) which can be configured as single-port RAM with a common address, enable, and clock control lines or two-port RAM with separate read and write control lines, address inputs and enable;
- register file and scratchpad RAM;
- FIFO memory for data buffering applications;
- 256-bit deep by 16-bit wide ROM with registered outputs, contents loaded during configuration;
- counters and sequencers.

5.6.3 Digital Signal Processing

As with other FPGA families, a DSP block is used which is an embedded block which can be configured into combination of the following functional units by selecting appropriate parameter values (Lattice 2014):

- a single 16×16-bit multiplier (generating 32-bit product output);
- two independent 8×8-bit multipliers (generating two independent 16-bit product output);
- a single 32-bit accumulator;
- two independent 16-bit accumulators;
- a single 32-bit adder/subtracter;
- two independent 16-bit adder/subtracters;
- a single 32-bit multiply-add/multiply-subtracter;
- two independent 16-bit multiply-adders/multiply-subtracters.

The key features of the Lattice iCE40isp FPGA family make it look similar to the other FPGA families, so the design process is similar to many of the other FPGAs. The key design focus is clearly the low-power design aspect.

5.7 MicroSemi RTG4 FPGA Family

Built on 65 nm process technology, Microsemi's RTG4 FPGA is a flash-based FPGA fabric with high-performance SERDES interfaces. Its key feature is its use of uPROM, a non-volatile flash memory, which uses the same flash technology as the FPGA configuration cells. uPROM is immune to memory upsets and has a total ionizing dose performance exceeding 100 krad, similar to the FPGA flash configuration cells. RTG4 devices have up to 374 kB of uPROM memory.

The RTG4 has up to 151,824 registers which are radiation hardened. Each logic element contains a four-input LUT with fast carry chains which can operate up to 300 MHz. There are also multiple embedded memory blocks and dedicated arithmetic units. A high-speed serial interface provides 3.125 Gbps native SERDES communication, while the DDR2/DDR3/ low-power double data rate (LPDDR) memory controllers provide high-speed memory interfaces. The RTG4 FPGA fabric is composed of four blocks, the logic module, LSRAM, uSRAM, and Mathblocks.

The RTG4 FPGAs are manufactured on a process with substantial reliability capability and are qualified to military standard (MIL-STD)-883 Class B. The devices are immune to SEU-induced changes in configuration, and thus no reconfiguration is required. The hardwired SEU-resistant flip-flops in the logic cells and in the Mathblocks avoid any data errors, due to radiation.

5.7.1 Programmable Logic Blocks

The logic module is the basic logic element and has the following advanced features:

- a fully permutable four-input LUT optimized for lowest power;
- a dedicated carry chain based on carry lookahead technique;
- a separate SEU-hardened flip-flop that can be used independently of the LUT where each flip-flop has its own synchronous reset while there is only one global asynchronous reset that drives all flip-flops.

The four-input LUTs can be configured to implement either a four-input combinatorial function or an arithmetic function, where the LUT output is XORed with the carry input to generate the sum output. Once again, the DSP system implementations will benefit from many of the techniques highlighted in Chapter 6. The availability of the carry lookahead technique will offer a performance gain as described in Chapter 3.

5.7.2 Memory

The RTG4 FPGA has a number of different memory flavors:

- **Dual-port LSRAM** is targeted for storing large amounts of data when implementing specific functions. Each block can store up to 24.5 kB and contains two independent data ports, A and B. The LSRAM block is synchronous and the data output ports have pipeline registers which have control signals that are independent of the SRAM's control signals.
- The **three-port uSRAM** has two read ports, A and B, which are independent of each other and can operate synchronously or asynchronously, and one write port, C, which

is always synchronous. The uSRAM block can store up to 1.5 kB and is primarily targeted for building embedded FIFOs or storing DSP coefficients.
- The **Non-volatile, flash uPROM memory** can be used for power-on initialization of RAMs and embedded IPs. The RTG4 devices have up to 374 kB of uPROM memory.

The uSRAMs are ideally suited to serve the needs of coefficient storage, and LSRAMs are used for data storage.

5.7.3 Mathblocks for DSP

The RTG4 comprises a number of custom 18×18-bit multiply-accumulate (MAC) blocks for efficient implementation of complex DSP algorithms. Each block has the following features:

- native 18×18-bit signed multiplications;
- dot product multiplication;
- built-in addition, subtraction, and accumulation units to combine multiplication results efficiently.

5.8 Design Stratregies for FPGA-based DSP Systems

Previous sections have described how the FPGA has evolved from an early interconnection technology to one involving processors and dedicated DSP technology features. The changes from a DSP perspective largely come down to changes in processing capabilities and memory.

5.8.1 DSP Processing Elements

As will be seen in the following sections, most FPGAs have leaned toward incorporation of a DSP block which is targeted mainly at fixed-point capability by allowing computations of the order of 9–10 bits. Accumulation capability is usually based around the need to accumulate numerous multiplications, as is the case for various forms of filtering and DSP transforms. The fact that the block is a dedicated silicon function means that it can run at reasonable clock rate, 500–600 MHz for the FPGAs considered here. Moreover, the capability to allow it to run at this higher speed is permitted by allowing pipelining to be employed.

This effectively means that this block should be treated as a fundamental building block for DSP systems and should be a core consideration when developing high-level models. For example, given the wordlength of typically 9 (or 18) bits, designers should treat increase in wordlength as a nonlinear function, i.e. when the design wordlength is increased from 9–10 bits, this represents a doubling in hardware. Thus these wordlength borderlines (9–10 and 18–19 bits) should be considered in the wordlength analysis.

Also the details of the actual DSP clocks should be taken into consideration in the detailed system modeling. Rather than just a multiplicative resource as in the earlier FPGAs, these DSP blocks have been developed with a lot of additive capability. In some cases, this can allow adder trees or linear chains of adders to be created. Again, this should be taken into consideration when developing the system description for the required algorithm.

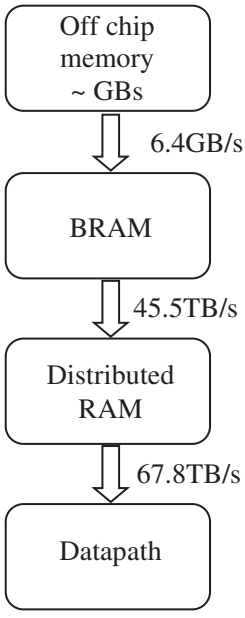

Figure 5.12 Bandwidth possibilities in FPGAs

5.8.2 Memory Organization

Compared to the computational processors, FPGAs are considered to have low levels of memory, and this has seen to be a major inhibitor for use in computational acceleration. However, as Figure 5.12 indicates for the Xilinx Virtex-7 FPGA, this is presented as a layered structure and thus can be thought of in the same way as cache memories in processors. The major positive aspect is the availability of high levels of bandwidth, as the figure highlights. Thus a strategy based on parallelism to take advantage of the large number of computational blocks would be highly beneficial.

As well as providing parallel storage opportunities, the availability of smaller memory blocks also suggests different implementation strategies for some systems, in terms of locally stored data, whether this is partially computed results or values for computations such as FFT coefficients. Moreover, the availability of small memory and fast adder logic as in many FPGA platforms has been the platform for distributed arithmetic and other approaches for fast DSP implementation.

5.8.3 Other FPGA-Based Design Guidelines

In addition to the summary on DSP units and memory organization, there are a number of other more detailed strategies that will clearly work well for the range of FPGAs presented in this chapter:

- Use pipelining at the circuit architecture level to improve performance. The availability of registers within the programmable hardware, and indeed the DSP units, means the pipelining can be deployed at no cost as the resources are readily available.

- Use the LUTs in the CLB functionality to implement efficient shift registers as RAM for storing filter coefficients. This comes from the capability implemented by most FPGA vendors to allow the LUTs to be used in in various modes.
- Develop DSP applications with a clear understanding of implementations of DSP block wordlengths so as to avoid using general-purpose interconnect. This involves restricting the usage to one column for highest performance and lowest power.
- Time multiplexing when resources are limited as the circuitry has been designed to operate at a specific speed and will be otherwise underutilized. This is because multiplexers are readily available and located near the LUTs and fast adder circuitry.
- It may also be advisable to use the CLB carry logic to implement small multipliers, adders, and counters rather than underutilize valuable DSP unit resources. This may be relevant in cases where FPGA resources are limited or the design is complex and could be accommodated in a smaller FPGA fabric.

5.9 Conclusions

The chapter has highlighted the variety of different technologies used for implementing DSP complex systems. These compare in terms of speed, power consumption and, of course, area (although this is a little difficult to ascertain for processor implementations). The chapter has taken a specific slant on programmability with regard to these technologies and, in particular, has highlighted how the underlying chip architecture can limit the performance. Indeed, the fact that it is possible to develop SoC architectures for ASIC and FPGA technologies is the key feature in achieving the high performance levels. It could be argued that the fact that FPGAs allow circuit architectures and are programmable are the dual factors that makes them so attractive for some system implementation problems.

Whilst the aim of the chapter has been to present different technologies and, in some cases, compare and contrast them, the reality is that modem DSP systems are now collections of these different platforms. Many companies are now offering complex DSP platforms comprising CPUs, DSP processors and embedded FPGA.

Bibliography

Altera Corp. 2015a Stratix V device overview. SV51001 2015.10.01. Available from www.altera.com (accessed February 16, 2016).

Altera Corp. 2015b A new FPGA architecture and leading-edge FinFET process technology promise to meet next generation system requirements. WP-01220-1.1. Available from www.altera.com (accessed February 16, 2016).

Altera Corp. 2015 Stratix 10: The most powerful, most efficient FPGA for signal processing. *Backgrounder*. Available from www.altera.com (accessed February 16, 2016).

Crockett LH, Elliot RA, Enderwitz MA, Stewart RW 2014 *The ZYNQ Book: Embedded Processing with the ARM Cortex-A9 on the Xilinx Zynq-7000 All Programmable SoC*. Strathclyde Academic Media, Glasgow.

Dobai R, Sekanina L 2013 Towards evolvable systems based on the Xilinx Zynq platform. In *Proc. IEEE Int. Conf. on in Evolvable Systems* pp. 89–95.

Lattice Semiconductor Corp. 2014 DSP function usage guide for iCE40 devices. Technical Note TN1295. Available from www.latticesemi.com (accessed February 16, 2016).

Lattice Semiconductor Corp. 2015 Memory usage guide for iCE40 devices. Technical Note TN1250. Available from www.latticesemi.com (accessed February 16, 2016).

Mindspeed 2012 Transcede® T33xx family of wireless application processors. Available from https://support.mindspeed.com/products /baseband-processors/transcede-3000, (accessed June 2, 2015).

Rose JS, Francis RJ, Lewis D, Chow P 1990 Architecture of field-programmable gate arrays: The effect of logic block functionality on area efficiency. *IEEE J. of Solid State Circuits*, 25(5), 1217–1225.

Siddiqui FM, Russell M, Bardak B, Woods R, Rafferty K 2014 IPPro: FPGA based image processing processor. In *Proc. IEEE Int. Conf. in Signal Processing Systems*, pp. 1–6.

Xilinx 2015a *UltraScale Architecture Configurable Logic Block*. User Guide UG574 v1.4. Available from www.xilinx.com (accessed February 16, 2016).

Xilinx 2015b *UltraScale Architecture Memory Resources*. User Guide UG573 v1.3. Available from www.xilinx.com (accessed February 16, 2016).

Xilinx 2015c *UltraScale Architecture DSP Slice*. User Guide UG579 v1.3. Available from www. xilinx.com (accessed February 16, 2016).

Yan H, Oruklu E 2014 Real-time traffic sign recognition based on Zynq FPGA and ARM SoCs. In *Proc. IEEE Int. Conf. on in Electro/Information Technology*, pp. 373–376.