Julian Rosker

Embedded Digital Signal Processing Techniques

Introduction

Embedded devices which have a non-trivial amount of signal analysis may suffer from bottlenecks at the signal processing step if the chosen device for signal processing is too low quality, rendering the device virtually useless [1]. Conversely, choosing too high a quality device for signal processing can increase final product costs to the point where the device cannot achieve market success [1]. This paper constitutes a brief summary of the current state of the different methods used for embedded digital signal processing applications with regards to relative strengths, weaknesses, and price.

Commercial Applications of Embedded Digital Signal Processing

The most notable commercial application of embedded digital signal processing is their widespread usage in mobile phones [2]. Within a mobile phone, a dedicated digital signal processor (DSP) is necessary to prevent the main processor from spending a large portion of its clock time on background processes [2]. Inside a mobile phone, the dedicated DSP is used for, among other tasks, encoding and decoding speech during phone calls, capturing and compressing videos, and processing game graphics [3]. As phone manufacturers rarely disclose the price of a phone's DSP (which are not commercially available), their prices can best be estimated via comparison with a commercially available DSP with similar specs. Most mobile phones would use a DSP with similar specs to the Texas Instruments (TI) 66AK2E05XABD25, which has a quad-core processor with a clock rate of 1.4GHz and can cost as little as $100.36 for bulk purchases [4]. A cheaper option would be the TI SMOMAPL138BGWTA3R, which is priced at $49.90 per unit but only boasts a 375MHz clock with two cores [5]. The latter option would be more suited for budget or low-power mobile phones. DSPs for mobile phones are mostly produced by Taiwan Semiconductor Manufacturing Company [6] or DSP Group [7], which both only list chip specifications but not individual prices, which is why estimation using TI's prices is necessary.

In the health industry, DSPs are used to reduce the cost and increase the quality of medical imaging [8]. MRIs, X-Rays, and Ultrasounds all benefit from an onboard DSP analyzing images in real time, allowing doctors to quickly diagnose potential problems [8]. Furthermore, the small size of DSPs make medical imaging techniques more portable, which is vital for providing healthcare to rural and low-income areas [8]. Of course, requirements for a DSP in a medical image processing vary widely based on intended usage, but the TI TMS320C6455BCTZA, a $263.91, 1.2GHz quad-core embedded DSP is representative of the type of device which would be needed for a portable ultrasound system [8][9].

In the financial field, digital signal processing is used to predict future market prices and react accordingly [10]. In a field where timing is everything, financial programmers often use field-programmable gate arrays (FPGAs) to speed up signal processing tasks]10]. FPGAs, which will be discussed more in a subsequent section, offer slight performance boosts in some cases for which traditional digital signal processors are not well suited [11]. The two predominant FPGA manufacturers are Xilinx and Altera, with the former holding a slightly larger market share (50% to 45%) [12]. Like digital signal processors, FPGAs have widely varying performance and costs. For instance, a Xilinx

XC7A200T-2FBG484I is priced at $256.10 each [13] while a low-performance Altera 10CL006YE144I7G is priced at $9.45 [14].

How Embedded Digital Signal Processing is Achieved

In embedded applications, there are two main ways to process signals: digital signal processors (DSPs) and field-programmable gate arrays (FPGAs) [15]. The former follows a standard Von Neumann architecture in which input in stored in short-term memory units and statements are processed sequentially [16]. This architecture can have lower hardware costs, but most of the savings come in the form of having a faster development time [16]. Since the technology is inherently easier to work with, test, and implement on a physical device, staff and technology reuse costs will be lower [16]. If the intended use is within a rapidly changing field, it may be desirable to sacrifice some performance or cost so that new iterations of a product can be released faster.

The alternative option to digital signal processors is field-programmable gate arrays (FPGAs), which offer potentially higher processing speeds but with high costs due to programming complexity [15]. The strength of FPGAs arises from the fact that each computation can be easily parallelized. In a digital signal processor, one can have at most one computation per clock cycle per core [15] [16]. For realistic (read: feasibly priced) embedded processors, it would be highly unlikely to have more than four cores. However, since at their core FPGAs are a collection of independent logic gates, a parallel-friendly process can have huge amounts of parallel tasks executing at once if ran on an FPGA [15].

Building Blocks for Implementation

Field-programmable gate arrays use three main elements: Look-Up-Tables, flip-flops, and a routing matrix [17]. A high-level design is written in a Hardware Description Language (HDL), and then is compiled to the target FPGA [17]. This design then coordinates the Loop-Up-Tables to provide functionality, such that an output for an arbitrary input can be “looked up” and then produced per the HDL [17]. Flip-flops and the routing matrix keep each step of the FPGA synchronized, ensuring that values are calculated in the correct order in the absence of a central processing unit [17]. A HDL compiler would also be needed to translate the programmer’s HDL to a form which the FPGA can understand and implement [18]. Fortunately for FPGA development, the actual implementation is trivial. FPGAs have I/O ports and, other than the compiled HDL and a small power supply, are self-contained units [18].

Digital signal processors use slightly more parts than FPGAs, which can be a hindrance for applications where conserving physical space is paramount. In addition to the processor, DSPs will require dedicated RAM and potentially a small amount of ROM [19]. This is in addition to a power supply which powers each of these components [19]. Programmers would likely write code for a digital signal processor in C++ or C, which requires a compiler to turn the code into machine language [19]. Both HDL and C/C++ have trusted open-source compilers, so the cost of the compiler itself should be a nonfactor in digital signal processing development.

References

[1] R. Baraniuk, "More Is Less: Signal Processing and the Data Deluge", *Science*, vol. 331, no. 6018, pp. 717-719, 2011.

[2] K. Markantonakis, “Embedded DSP Devices ,“ in *Secure Smart Embedded Devices, Platforms and Applications*, Ed. New York, Springer, 2013, ch. 2, pp. 28-40.

[3] S. Jadhav, "Use of DSP for wireless and mobile Communication", Ph.D, Indian Institute of Technology, 2017.

[4] Texas Instruments, “66AK2E0x Multicore DSP+ARM KeyStone II System-on-Chip (SoC),” 66AK2E05XABD25 datasheet, Nov. 2012 [Revised Mar. 2015].

[5] Texas Instruments, “SMOMAPL138B Low-Power Applications Processor,” SMOMAPL138BGWTA3R datasheet, Jan. 2011 [Revised Jul. 2013].

[6] I. Cutress, "Apple 2017: The iPhone X (Ten) Announced", *Anandtech.com*, 2017. [Online]. Available: https://www.anandtech.com/show/11835/apple-2017-the-iphone-x-ten. [Accessed: Oct. 23 2017].

[7] "DSP Group to see revenue boost from Samsung S7 chip win", *Reuters.*, 2017. [Online]. Available: http://www.reuters.com/article/us-dsp-group-samsung-elec-chip/dsp-group-to-see-revenue-boost-from-samsung-s7-chip-win-idUSKCN0W91AR. [Accessed: Oct. 22 2017].

[8] M. Nadeski and A. Gatherer, "See the difference: DSPs in medical imaging", Texas Instruments, 2017.

[9] Texas Instruments, “TMS320C6455 Fixed-Point Digital Signal Processor,” TMS320C6455 datasheet, May 2005 [Revised Mar. 2012].

[10] J. Lockwood, et al, "A Low-Latency Library in FPGA Hardware for High-Frequency Trading (HFT)", *2012 IEEE 20th Annual Symposium on High-Performance Interconnects*, 2012.

[11] J. Ehlers, *Cybernetic analysis for stocks and futures*. Hoboken, N.J.: Wiley, 2013.

[12] K. Morris, "Xilinx vs. Altera – EEJournal", *Eejournal.com*, 2014. [Online]. Available: https://www.eejournal.com/article/20140225-rivalry/. [Accessed: October 23 2017].

[13] XILINX, “7 Series FPGAs Data Sheet: Overview,” XC7A200T-2FBG484I datasheet, Aug. 2017.

[14] Intel, “Intel® Cyclone® 10 LP Device Overview,” Altera 10CL006YE144I7G datasheet, May. 2017.

[15] Sepulveda, C., Munoz, J., Espinoza, J., Figueroa, M. and Baier, C. (2013). “FPGA v/s DSP Performance Comparison for a VSC-Based STATCOM Control Application”. *IEEE Transactions on Industrial Informatics*, 9(3), pp.1351-1360.

[16] K. Keutzer, "http://bwrcs.eecs.berkeley.edu/Classes/CS252/Notes/Lec09-DSP.pdf", Berkeley, 2000.

[17] "How does an FPGA work?", *Embeddedmicro.com*, 2017. [Online]. Available: https://embeddedmicro.com/tutorials/mojo-fpga-beginners-guide/how-does-an-fpga-work. [Accessed: October 23 2017].

[18] "Design through Verilog HDL", *Choice Reviews Online*, vol. 41, no. 08, 2004.

[19] "Digital Signal Processing Software and Development Tools Selection Guide", 2009. [Online]. Available: http://www.ti.com/lit/ml/sprb191b/sprb191b.pdf. [Accessed: October 23 2017].