# Technical Review Paper Evaluation Form

(attach this form as the cover page for your report)

**Student Name:** Sriram Pulavarty

**Project Advisor:** Professor Jennifer Hasler

**Team Name:** Music Electronics

**Team Members:** Justin Kelley, Sriram Pulavarty, Harrison Zhang, Kristyn DiGiovanni, Jongheon Park, Chris Walds, and Yewon Kim

_______ / 30 Technical Content

- Current state-of-the-art and commercial products
- Underlying technology
- Implementation of the technology
- Overall quality of the technical summary

_______ / 30 Use of Technical Reference Sources

- Appropriate number of sources (at least six)
- Sufficient number of source types (at least four)
- Quality of the sources
- Appropriate citations in body of text
- Reference list in proper format

_______ / 40 Effectiveness of Writing, Organization, and Development of Content

- Introductory paragraph
- Clear flow of information
- Organization
- Grammar, spelling, punctuation
- Style, readability, audience appropriateness, conformance to standards

**_______ / 100 Total - Technical Review Paper**

**Field Programmable Analog Arrays for Music Synthesis**

## Introduction

Due to their tendency to produce distortion as well as subtle variations in waveform shape, frequency, and amplitude; analog music synthesizers are often considered to produce more natural, aurally pleasing sounds than their digital counterparts [1]. However, general purpose analog computers can be expensive and difficult to program for synthesis applications. As a result, most analog synthesizers eschew low-level analog computing blocks for high-level analog modules that implement user-defined functions, sacrificing programmability for convenience [2]. Field Programmable Analog Arrays (FPAAs) – blocks of analog components that can be connected in an arbitrary fashion – aim to eliminate this issue by providing an energy efficient, easily programmable alternative to modular synthesis. This paper reviews some commercially available FPAAs, details the state of FPAA technology, and provides an outline of their basic operation with regard to music synthesis.

## Commercial Availability and Applications of FPAAs

Although many universities, such as the Georgia Institute of Technology, the University of Toronto, and West Virginia University have done research on FPAAs, high production costs have limited the technology's commercial availability. However, there are currently two corporations at the forefront of FPAA production: Anadigm and Aspinity [3]. In addition to a variety of specialized analog modules, such as its SonicMaster and RangeMaster ranges of products, Anadigm sells general purpose FPAAs as dpASP (dynamically programmable Analog Signal Processors) devices. Its newest model, the AN231E04 FPAA, is based primarily on switched capacitor technology and is priced at $8.46 [4]. The AN231E04 operates on a 3.3 V power supply and contains a clock generator; a 256 byte look-up table; and four Configurable Analog Blocks (CABs) that each contain two operational amplifiers, a comparator, banks of programmable capacitors, and clocking and routing resources. The device can be programmed with Anadigm's custom AnadigmDesigner 2 software and is targeted at analog signal processing, RFID baseband filtering, and real-time control of analog peripherals [5] – [6]. Aspinity's product range is more limited. Its RAMP (Reconfigurable Analog Modular Processor) FPAA incorporates 80 logic blocks in the form of both digital Configurable Logic Blocks (CLBs) and analog Computational Analog Blocks (CABs). RAMP's CLBs contain flip flops and look-up tables, while its CABs rely on floating gate transistors in addition to passive elements such as resistors and capacitors, enabling it to achieve better low power performance than its competitors [7]. RAMP is targeted specifically at wireless sensing applications, and Aspinity offers custom versions of the product for acoustic-event detection and vibration monitoring [8].

## Technology

### Hardware Functionality

While there are a multitude of different FPAA designs, most FPAAs consist of arrays of CABs (Computational Analog Blocks), a switch matrix that defines the interconnections between the CABs, and a programmer unit that can control each device on the FPAA [2], [9]. The switch matrix and the CABs can be made from a variety of different technologies. Floating gate technology is one option. In floating gate technology, the traditional MOSFET structure is modified so that an electrically isolated gate lies between the control gate and the substrate layer, ensuring that charge can only be added or removed via quantum mechanical procedures, such as hot electron injection or tunneling. This allows the transistors to be programmed to very specific voltages for long periods of time so that they can finely control voltage paths through the routing "fabric" of the FPAA by turning on, turning off, or even switching to other voltage levels [10]. By controlling voltage paths in such a manner or by simply varying their output voltages, floating gate transistors can serve as the basis for an FPAA's switch matrix or (when combined with passive components such as resistors and capacitors) CABs, respectively. They can also be used to enhance an FPAA's reconfigurability; a row of floating gate transmission gates set by a shift register can be placed between each CAB and its interconnect to allow a user to quickly scan data from various CABs by shifting through different lines if necessary [9]. However, floating gate transistors are not the only technology that can be used in FPAAs, though they are generally the most energy efficient. Other companies, such as Anadigm, have used a combination of switched capacitors and operational amplifiers to build the switch matrices and CABs on their FPAAs [3].

### Programming FPAAs

The benefits of a complex analog circuit are often irrelevant if it is not reconfigurable. As a result, the programmability of an FPAA is often a primary concern. Different organizations have come up with different ways of programming FPAAs, but nearly all of them include some level of digital circuitry to make the process easier. Anadigm, for instance, includes SRAM cells in each CAB on its AN231E04 device so that users can easily program it with Anadigm's proprietary AnadigmDesigner 2 software, which lists gain, rectification, filtering, summation, differentiation, integration, multiplication, transfer function, comparator, sample and hold, oscillator sine, and voltage functions for each block [3]. A research group at the Georgia Institute of Technology took a more radical approach, incorporating an entire microprocessor onto their FPAA along with a 16k x 16 SRAM block, an array of memory mapped registers, and a set of digital Configurable Logic Blocks (CLBs) interspersed with the FPAA's CABs [9]. This design allows for a complex, multilayered software interface between the user and the device. To program the Georgia Institute of Technology's FPAA, the user interacts directly with a high end design environment built in Xcos that compiles into distinct analog, digital, and assembly components. The analog and digital components are converted into netlists with blif (Berkeley Library Interface), which are then compiled to a switch list for the FPAA's program unit with an open source tool called x2c. Meanwhile, the assembly component of the software is compiled into hex code, where it is also delivered to the FPAA's program unit [11]. Such a process allows the user a great deal of control over both the analog and digital components of the device while maintaining a level of convenience.

**Implementing Music Synthesis with FPAAs**

FPAAs can be used to synthesize music by creating blocks that act as voltage controlled filters (VCFs), oscillators (VCOs), or amplifiers (VCAs) [2]. These blocks can be formed from various components that are contained in CABs, such as current starved inverters, Gilbert Multipliers, and transistor ladder filters. One form of synthesis, called subtractive synthesis, combines VCFs, VCOs, and VCAs to create a harmonically rich signal before "subtracting" certain amplitudes and frequencies to produce a desired sound. The first step in subtractive synthesis is to generate a periodic waveform with the VCO. The frequency of the waveform can be controlled with input Control Voltages (CVs) until the desired value is obtained. Once the waveform has been generated, it is fed into the VCA to create an amplitude envelope for the signal. The amplitude envelope can also be controlled by CVs input into the VCA until the desired envelope is obtained. One particular envelope, called ADSR (attack-decay-sustain-release), is commonly used for subtractive synthesis. After the waveform passes through the VCA, it is input into the VCF, whose cutoff frequency is determined by a CV. The cutoff frequency can be changed to determine which frequencies to "subtract", and sometimes the CV may be varied over time in order to produce different sounds [12]. Ultimately, music synthesis with FPAAs is largely dependent on the input CVs for the VCFs, VCOs, and VCAs on the device, and much of the process can be focused on varying these to achieve the best sound.

[1] M. Doty, "What's Behind The Resurgence of Analog Synthesizers?", *Performer Mag*, July 14, 2016. [Online]. Available: https://performermag.com/best-instruments/best-music-keyboards-synth/resurgence-of-analog-synthesizers/. [Accessed Sept. 28, 2019]

[2] V. Lazzarini and J. Timoney, "The Analogue Computer as a Voltage-Controlled Synthesiser", *Maynooth University*. April 25, 2019. [Online]. Available: https://arxiv.org/pdf/1904.10763v2.pdf [Accessed Sept. 28, 2019]

[3] T. Leddicker, "Sensors: Moving the analog-digital boundary", *Evaluation Engineering*, May 19, 2016. [Online]. Available: https://www.evaluationengineering.com/instrumentation/article/13013798/sensors-moving-the-analogdigital-boundary [Accessed Sept. 28, 2019]

[4] Anadigm, Inc., "The Anadigm Product Range", *Anadigm, Inc.*, [Online]. Available: https://www.anadigm.com/products.asp [Accessed Sept. 28, 2019]

[5] Anadigm Product Staff, *AN231E04 Datasheet Rev 1.2*, Anadigm, Inc., 2014. [Online]. Available: https://www.anadigm.com/_doc/DS231000-U001.pdf [Accessed Sept. 28, 2019]

[6] Anadigm Product Staff, *AnadigmApex dpASP Family User Manual*, Anadigm, Inc., [Online]. Available: https://www.anadigm.com/_doc/UM000231-U001.pdf [Accessed Sept. 28, 2019]

[7] B. Rumberg, D. Graham, S. Clites, B. Kelly, M. Navidi, A. Dilello, and V. Kulathumani, "RAMP: Accelerating Wireless Sensor Hardware Design with a Reconfigurable Analog/Mixed-Signal Platform", *International Conference on Information Processing in Sensor Networks*. April 14 – 16, 2015. [Online]. Available: http://community.wvu.edu/~dwgraham/publications/papers/ramp_ipsn2015.pdf [Accessed Sept. 28, 2019]

[8] Aspinity, Inc. "Applications", *Aspinity, Inc.*, [Online]. Available: https://www.aspinity.com/applications [Accessed Sept. 28, 2019]

[9] S. George, S. Kim, S. Shah, J. Hasler, M. Collins, F. Adil, R. Wunderlich, S. Nease, and S. Ramakrishnan "A Programmable and Configurable Mixed-Mode FPAA SOC," IEEE Transactions on VLSI, January 2016. [Online]. Available: http://hasler.ece.gatech.edu/SoCFPAA/RASP30_SoCFPAA.pdf [Accessed Sept. 29, 2019]

[10] S. Brink, J. Hasler, and R. Wunderlich, "Adaptive Floating-Gate Circuit Enabled Large-Scale FPAA," IEEE Transactions on VLSI Systems, 2014. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6705658 [Accessed Sept. 29, 2019]

[11] M. Collins, J. Hasler, and S. George, "An Open-Source Tool Set Enabling Analog-Digital-Software Co-Design," Journal of Low Power Electronics and Applications, January 2016. [Online]. Available: http://hasler.ece.gatech.edu/Published_papers/FPAA_Papers/Highlevel_tools_jlpea_February2016.pdf [Accessed Sept. 29, 2019]

[12] S. Nease. "Neural and Analog Computation on Reconfigurable Mixed-Signal Platforms", Ph. D. dissertation, Georgia Institute of Technology, GA, 2014.