DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

ECE1
(B1) Polynomial Evaluation Accelerator (PEA)

Hailey Chawla, Emma Gass, May Lee

A. JAMES CLARK SCHOOL OF ENGINEERING

Problem Definition

Modern digital signal processing and embedded computing systems frequently require repeated polynomial evaluations over large data streams. General-purpose processors handle these computations inefficiently due to instruction fetch overhead, memory latency, and lack of dedicated arithmetic pipelines. This project addresses the need for a dedicated hardware accelerator that can evaluate polynomials rapidly and with minimal resource overhead. The Polynomial Evaluation Accelerator (PEA) is designed to interface with a streaming dataflow architecture, accepting instructions and data through FIFO buffers, and producing results with low latency. The PEA supports four instructions:

- STP - Store Polynomial. Loads a set of coefficients into one of the 8 coefficient vectors stored inside the PEA.
- EVP - Evaluate Polynomial. Evaluates a stored polynomial at a single input argument *x* and outputs the result.
- EVB - Evaluate Block. Evaluates a stored polynomial repeatedly over a block of b input arguments, essentially a shorthand for running EVP *b* times with the same polynomial.
- RST - Reset. Clears all 8 stored coefficient vectors back to an empty/invalid state.

These instructions enable flexible polynomial storage, evaluation, and block processing. The design is implemented both as a software actor in LIDE-C and as a synthesizable hardware design in Verilog, allowing direct comparison of functional correctness and hardware performance trade-offs.

Design Calculations & Analysis

Hardware design requires balancing two competing objectives: **resource usage and performance**. No single design minimizes both simultaneously. To explore this tradeoff, we implemented three Pareto optimal designs. A design is Pareto optimal when no other design simultaneously achieves better performance and lower resource usage. To go faster you must spend more hardware, and to save hardware you must accept slower execution. Neither extreme is wrong as they simply serve different priorities.

For a polynomial of degree N evaluated over a block of b arguments:

- Design 1 requires 10 times b cycles. Each of the b evaluations requires N sequential Horner steps, one per cycle.
- Design 2 requires 5b cycles using a two-stage pipelined Horner implementation with 2 parallel DSP multipliers, running at a higher clock frequency than Design 1.
- Design 3 requires 8 plus b cycles. The 8-cycle startup cost builds the power chain x through x^9, after which each subsequent evaluation in the block requires only one additional cycle since all terms are computed in parallel.

For large block sizes Design 3 dominates in throughput. For small block sizes or single evaluations, Design 2 offers a compelling tradeoff. Design 1 remains the right choice when FPGA resources are tightly constrained.

Pareto Optimality.
Design 1 cannot be beaten on resource usage. Design 3 cannot be beaten on throughput for large block sizes. Design 2 occupies the middle point where doubling the DSP count yields a meaningful clock frequency improvement for modest additional resource cost. All three designs are Pareto optimal as each is the best available option at its point on the resource-performance curve.

A dominated design would be one that is both slower AND more resource-hungry than another design but there would be no reason to ever choose it. All three of our designs avoid this by each being the best option at their respective point on the resource-performance curve. The Pareto frontier is the curve connecting these three optimal points. A designer chooses where on this curve to land based on their system constraints. If chip area and resource usage are limited, Design 1 is the right choice. If maximum clock frequency is the priority, Design 2 is preferred. If the application involves evaluating large blocks of inputs where startup cost is amortized across many evaluations, Design 3 becomes advantageous. Design 2 offers a compelling middle ground for general use, achieving the highest frequency with moderate resource cost.

Final Design

The **PEA** is implemented as a two-component system with a software actor in **LIDE-C** and a Synthesizable Verilog hardware implementation using the **LIDE-V** framework.

Software Component (LIDE-C):
The LIDE-C actor follows Core Functional Dataflow (CFDF) semantics with five modes: PROCESS, STP, EVP, EVB, and RST. The actor begins in PROCESS mode, decoding a 16-bit instruction word and transitioning to the appropriate mode based on the 2-bit opcode field. Each mode has fixed data consumption and production rates as required by CFDF semantics. The instruction word encodes a 2-bit opcode, a 3-bit CV address, and a 5-bit operand field. Error detection is performed at each transition, producing dedicated status codes for conditions such as uninitialized CVs, out-of-range degree values, and invalid block sizes. The implementation was validated with three unit tests per instruction and five multi-instruction stream tests.

Hardware Component (LIDE-V - Verilog):
The PEA maintains eight coefficient vectors supporting polynomials up to degree 10, with 16-bit signed two's complement inputs and 32-bit signed results. Each Verilog design follows the same LIDE-V top level structure with IDLE, FIRING-START, and FIRING-WAIT states in the outer FSM, and a nested control FSM handling all instruction modes internally. Three Pareto optimal designs were implemented and synthesized using Vivado:

- Design 1 → Single multiply-accumulate unit implementing Horner's method sequentially. One Horner step is computed per clock cycle, requiring N cycles to evaluate a degree-N polynomial. This design minimizes hardware resource usage and represents the lowest cost point on the Pareto frontier.
- Design 2 → Two-stage pipeline with two parallel DSP multipliers implementing Horner's method. Stage 1 computes the accumulator multiplied by x and registers the result. Stage 2 takes the registered output and adds the next coefficient. Because a pipeline register separates the two stages, Vivado infers them as independent DSP blocks. A degree-N polynomial requires N pipeline cycles, but each cycle has a shorter critical path allowing a higher maximum clock frequency than Design 1.
- Design 3 → 19 parallel multipliers to evaluate all polynomial terms simultaneously using direct evaluation rather than Horner's method. Nine multipliers form a power chain computing x, x^2, x^3 up to x^9, and ten multipliers compute each term simultaneously. This achieves evaluation in 8 plus b cycles for a block of size b, making it the fastest design at the cost of the highest resource usage.

Current state	Condition	Next state	Action
IDLE	ctrl FIFO empty	IDLE	Stall (enable = false)
IDLE	ctrl FIFO not empty	FETCH	Dequeue instruction word
FETCH	always	DECODE	Latch opcode, A, N/b fields
DECODE	opcode = STP	STP_LOAD	Init coeff counter = 0
DECODE	opcode = EVP	EVP_EXEC	Read x from data FIFO
DECODE	opcode = EVB	EVB_EXEC	Init block counter = 0
DECODE	opcode = RST	RST_EXEC	Clear all CV valid flags
STP_LOAD	coeff_cnt ≤ N, data FIFO not empty	STP_LOAD	Read coeff, store in CV[A][cnt++]
STP_LOAD	coeff_cnt > N	OUTPUT	Mark CV[A] valid, set degree = N
EVP_EXEC	evaluation complete	OUTPUT	Compute P_A(x), latch result
EVB_EXEC	blk_cnt < b	EVB_EXEC	Read y, evaluate, output, blk_cnt++
EVB_EXEC	blk_cnt = b	IDLE	All b evaluations done
RST_EXEC	always	IDLE	All CVs invalidated, no output
OUTPUT	both output FIFOs not full	IDLE	Write result + status to FIFOs

Figure 1: Control FSM State Transition Table

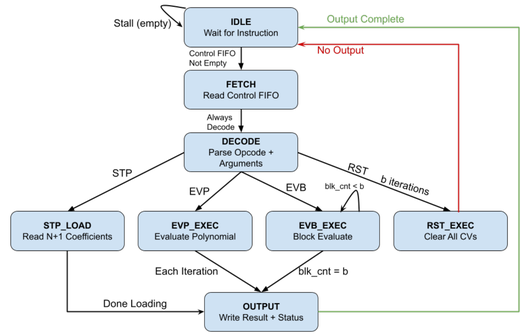

Figure 2: Transition Diagram for the PEA LIDE-C Actor

Prototype & Test Results

Metric	Design 1	Design 2	Design 3
Look Up Tables (LUTs)	653	1141	997
Flip-Flops	1622	1688	2818
DSP Blocks	3	3	19
Max Frequency	127 MHz	228 MHz	210 MHz

Figure 3: Table of Synthesis and Implementation Results

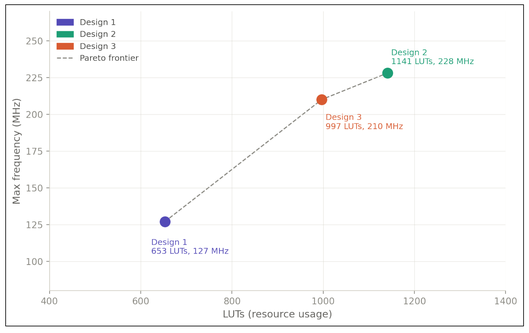

Figure 4: Pareto Curve Comparison of Designs

The synthesis results reveal a more nuanced picture than the original design intent suggested. When plotting LUTs against maximum frequency, all three designs occupy distinct and non-dominated positions. Design 1 sits at the minimum resource corner with 653 LUTs and 127 MHz. Design 2 sits at the maximum performance corner with 1,141 LUTs and 228 MHz. Design 3 sits between them at 997 LUTs and 210 MHz, using fewer resources than Design 2 but achieving lower performance.

This means the Pareto frontier runs from Design 1 → Design 3 → Design 2 in order of increasing resource usage and increasing performance. No design is dominated because each one is either cheaper or faster than the others, meaning you cannot find a design that beats any of them on both axes simultaneously.

For the PEA application, Design 2 is the best choice when throughput matters most since it achieves the highest clock frequency. Design 1 is the best choice when the FPGA area is constrained. Design 3 offers a middle ground with high parallelism and good frequency, making it most advantageous for very large block sizes where the 8-cycle startup cost is amortized across many evaluations.

The Pareto curve confirms that all three designs are valid optimal choices depending on the system constraints, and no design is wasteful or redundant.