# Programming FPGAs for Economics:

# An Introduction to Electrical Engineering Economics

## Replication Package
## PARSER

Bhagath Cheela[*]          André DeHon[†]

Jesús Fernández-Villaverde[‡]          Alessandro Peri[§]

September 25, 2024

# 1   Introduction

This document serves as `parser` file for replicating all tables presented in the paper "*Programming FPGAs for Economics: An Introduction to Electrical Engineering Economic*", by Bhagath Cheela, André DeHon, Jesús Fernández-Villaverde and Alessandro Peri.

---

[*]Department of Electrical and Systems Engineering, University of Pennsylvania, cheelabhagath@gmail.com
[†]Department of Electrical and Systems Engineering, University of Pennsylvania, andre@acm.org
[‡]Department of Economics, University of Pennsylvania, jesusfv@econ.upenn.edu
[§]Department of Economics, University of Colorado, Boulder, alessandro.peri@colorado.edu

# 2 Tables

Table 1: Calibrated Parameters

| | | |
|---|---|---|
| $\alpha$ | 0.36 | Output capital share |
| $\beta$ | 0.99 | Quarterly discount factor |
| $\gamma$ | 1 | Relative risk aversion coefficient |
| $\delta$ | 0.025 | Quarterly depreciation rate |
| $\mu$ | 0.15 | Unemployment benefits in terms of wages |
| $\bar{l}$ | 1/0.9 | Time endowment |
| $\Delta_A$ | 0.01 | Aggregate productivity shock size |

Table 2: Benchmarking the CPU: Alternative Search Algorithms

| | **Linear Search** | **Binary Search** | **Jump Search** |
|---|---|---|---|
| *Solution Time* | 73657.8 | 38392.0 | 28452.5 |
| *Speedup* | - | 1.92 | 2.59 |

*Note:* Solution time (in seconds) and speedups of alternative interpolation interval search algorithms. Speedups are computed relative to the linear search algorithm. Results are obtained by solving 1,200 baseline economies sequentially using a single core instance (m5n.large).

Table 3: Efficiency Gains and Implementation Costs of FPGA Acceleration

Panel A: Efficiency Gains of FPGA Acceleration

| CPU-cores | Speedup | | | Relative Costs (%) | | | Energy (%) | | |
|---|---|---|---|---|---|---|---|---|---|
| | *FPGAs* | | | *FPGAs* | | | *FPGAs* | | |
| | *1* | *2* | *8* | *1* | *2* | *8* | *1* | *2* | *8* |
| *1* | 68.54 | 137.09 | 548.56 | 20.23 | 20.23 | 20.22 | 6.02 | 6.02 | 6.02 |
| *8* | 8.80 | 17.61 | 70.46 | 19.69 | 19.69 | 19.68 | 5.86 | 5.86 | 5.85 |
| *48* | 1.48 | 2.96 | 11.83 | 19.55 | 19.55 | 19.54 | 5.82 | 5.82 | 5.81 |

Panel B: Implementation Costs of FPGA Acceleration

| | *Kernel* | | *Non-kernel* | |
|---|---|---|---|---|
| | Number | Percent (%) | Number | Percent (%) |
| *Extra Lines of Code* | 75 | 5.37 | 128 | 51 |

*Note:* Panel A reports speedups provided by the FPGA and cost and energy usage of the FPGA relative to the CPU. The results are obtained by solving 1,200 baseline economies using AWS instances connected to 1, 2, and 8 FPGAs and using open-MPI parallelization on AWS instances with 1, 8, and 48 cores (rows). Speedup is obtained by dividing the total solution time in the CPU by that in the FPGA. Relative costs and energy are calculated using on-demand AWS prices and total energy consumption, and reported as FPGA usage as a percent of CPU usage. Table A.4 in Appendix **??** reports the details. Panel B estimates implementation costs for both kernel and non-kernel segments of our codebase by reporting the extra lines of code required by the HLS-enhanced C code when compared to standard C code designed to be executed on the CPU using Open MPI.

## Table 4: Single-Kernel FPGA vs. Single CPU Core

### Panel A: Benchmark Model, $\{N_k, N_M\} = \{100, 4\}$

| $FPGA$-Time(sec) | $CPU$-Time(sec) | $Speedup$(x) | $Relative\ Costs$(%) | $Energy$(%) |
|---|---|---|---|---|
| 0.84 | 23.71 | 28.38 | 48.86 | 7.49 |

### Panel B: Speedup across Grid Sizes

| Aggregate Capital, $N_M$ | 4 | | | | 8 | | |
|---|---|---|---|---|---|---|---|
| Individual Capital, $N_k$ | *100* | *200* | *300* | | *100* | *200* | *300* |
| *Speedup (x)* | 28.38 | 34.41 | 34.31 | | 27.81 | 31.05 | 32.06 |
| *Relative Costs (%)* | 48.86 | 40.30 | 40.41 | | 49.85 | 44.65 | 43.26 |
| *Energy (%)* | 7.49 | 6.18 | 6.19 | | 7.64 | 6.84 | 6.63 |

*Note:* Figures are obtained by comparing the solution of 1,200 economies using AWS instances connected to 1 FPGA and sequential CPU execution on a single core. Panel A focuses on the benchmark economy, $\{N_k, N_M\} = \{100, 4\}$. Columns 1-2 detail the average solution time (in seconds) to compute the benchmark economy in a single-kernel, single-device FPGA (f1.2xlarge), and a single-core instance (m5n.large), respectively. Columns 3-5 display the efficiency gains of FPGA acceleration in terms of speedup, costs (in percent), and energy savings (in percent), computed as described in Table 3. The FPGA average power consumption on a single-kernel design is 17 watts. Panel B studies how speedup, relative costs, and energy consumption vary with the size (columns) of the individual household capital holdings grid ($N_k$) and aggregate capital grid ($N_M$).

Table 5: Speedup Gains: Acceleration Channels Accounting

|  | Baseline | Pipelining | Data Parallelism | |
|  |  |  | Within Economy | Across Economies |
| --- | --- | --- | --- | --- |
| $\dfrac{\text{Single-core Execution}}{\text{FPGA Solution}}$ | 0.21 | 6.94 | 28.38 | 68.54 |
| *CL Resources Utilization (%)* |  |  |  |  |
| BRAM | 6.01 | 7.14 | 21.31 | 44.29 |
| DSP | 7.75 | 9.68 | 31.13 | 55.32 |
| Registers | 3.99 | 5.12 | 12.00 | 25.71 |
| LUT | 5.96 | 9.20 | 25.21 | 57.03 |
| URAM | 5.50 | 5.50 | 5.38 | 16.50 |

*Note:* Column 1 reports the speedup for a kernel design where all acceleration channels are switched off (baseline). Columns 2-4 report the speedup associated with implementing efficient pipelines (Column 2), introducing data parallelism in the kernel design (Column 3), and instantiating three kernels in the same FPGA (Column 4). The speedup (row 1) is computed by dividing the total solution time in the one-core CPU by the solution time in the FPGA. The acceleration in Columns 1-3 is performed using a single-kernel, single-device FPGA (f1.2xlarge), where Column 4 coincides with the single-kernel design. The acceleration in Column 4 is performed by deploying the three-kernel design in parallel across the three SLRs in a single FPGA (f1.2xlarge). Averages are computed over 1,200 economies, except for the Baseline and Pipeline designs, which for cost considerations are computed over 120 economies. Resources are measured (using Xilinx Vivado) as a percentage of the Xilinx VU9P FPGA's resources utilized by AWS images associated with the different designs (columns). *Available Resources: BRAM (1,680), DSP (5,640), Registers (1,790,400), LUTs (895 thousand), URAM (800).*

# A  Online Appendix Tables

Table A.1: **List of Abbreviations**

| | | |
|---|---|---|
| ALM | Aggregate Law of Motion | Algorithm stage |
| AFI | Amazon FPGA Image | CL design implemented on AWS FPGAs |
| AWS | Amazon Web Services | Cloud service |
| .AWSXCLBIN | FPGA executable | Executable to be run on AWS FPGA |
| BRAM | Block RAM | Local memory |
| CL | Custom logic | FPGA logical units |
| CPU | Central processing unit | - |
| DRAM | Dynamic random access memory | Global memory |
| DSP | Digital signal processing unit | Accumulator unit |
| FPGA | Field-programmable gate array | Custom accelerator |
| GPU | Graphics processing unit | Graphics accelerator |
| HLS | High level synthesis | Compiler-based hardware design |
| IEEE754 | Double-precision floating-point standard | Floating-point standard |
| IHP | Individual Household Problem | Algorithm stage |
| II | Initiation Interval | |
| LUT | Lookup table | Logical units available for CL design |
| OpenCL | Open Computing Language | https://www.khronos.org/opencl |
| Open MPI | Open message passing interface | https://www.open-mpi.org |
| PCIe | Peripheral Component Interconnect Express | Bus-connections with host |
| SLR | Super Logic Region | FPGA CL regions |
| URAM | Ultra RAM | Local memory |
| Xilinx VU9 | FPGA on AWS | - |

Table A.2: Technical Specifications

| AWS Instance | Cores | FPGAs | Pricing ($/hour) | Memory (GiB) |
|---|---|---|---|---|
| m5n.large | 1 | - | 0.119 | 8 |
| m5n.4xlarge | 8 | - | 0.952 | 64 |
| m5n.24xlarge | 48 | - | 5.712 | 384 |
| f1.2xlarge | 4 | 1 | 1.650 | 122 |
| f1.4xlarge | 8 | 2 | 3.300 | 244 |
| f1.16xlarge | 32 | 8 | 13.200 | 976 |

*Note:* Hardware architecture and AWS cloud pricing (Columns 2-5) for deployed AWS instances (Column 1). The column marked Cores reports the number of physical cores. The column marked FPGAs reports the number of connected FPGA chips (f1 instances only). The column marked Pricing denotes the AWS *On Demand* Pricing per instance per hour as of September 2021. Memory is measured in Gigabytes. *Source:* AWS instances, AWS specs.

Table A.3: FPGA Designs Performance and Resource Utilization by Grid Size

| | Three-Kernel | Single-Kernel | | | | | |
|---|---|---|---|---|---|---|---|
| Aggr. Capital | *4* | *4* | | | *8* | | |
| Indiv. Capital | *100* | *100* | *200* | *300* | *100* | *200* | *300* |
| Time (s) | 415.14 | 1002.62 | 1482.11 | 2245.56 | 2579.66 | 4627.80 | 7147.36 |
| Cost ($) | 0.19 | 0.46 | 0.68 | 1.03 | 1.18 | 2.12 | 3.28 |
| Energy (J) | 13699.54 | 17044.46 | 25195.90 | 38174.60 | 43854.19 | 78672.63 | 121505.20 |
| BRAM(%) | 44.29 | 21.31 | 27.32 | 33.10 | 27.32 | 37.92 | 47.26 |
| DSP(%) | 55.32 | 31.13 | 31.13 | 31.13 | 31.31 | 31.31 | 31.31 |
| Registers(%) | 25.71 | 12.00 | 12.00 | 12.12 | 12.06 | 12.17 | 12.26 |
| LUT(%) | 57.03 | 25.21 | 25.97 | 26.56 | 25.43 | 26.18 | 26.74 |
| URAM(%) | 16.50 | 5.38 | 5.38 | 5.38 | 5.38 | 5.38 | 5.38 |

*Note:* Solution time (in seconds), cost (in USD), energy (in joules) and FPGA resources (rows) across hardware designs (three- and single-kernel) and grid sizes on individual capital $N_k = \{100, 200, 300\}$ and aggregate capital $N_M = \{4, 8\}$ (columns). Time performance is measured in seconds required to solve 1,200 baseline economies on a single FPGA (f1.2xlarge) across the different hardware designs and grid sizes (columns). Resources are measured (using Xilinx Vivado) as a percentage of Xilinx VU9P FPGA's resources utilized by AWS images associated with the different hardware designs and grid sizes (columns). *Available Resources:* BRAM (1,680), DSP (5,640), Registers (1,790,400), LUTs (895 thousand), URAM (800). Available resources are lower than total resources because they exclude resources utilized by the AWS shell that are not available for `CL` design.

Table A.4: Performance Comparison

| | CPU cores | | | FPGA devices | | |
|---|---|---|---|---|---|---|
| N. | 1 | 8 | 48 | 1 | 2 | 8 |
| Exec Time (s) | 28464.55 | 3656.52 | 613.81 | 431.60 | 223.40 | 69.51 |
| Init Time (s) | 0.36 | 0.04 | 0.01 | 0.81 | 0.67 | 0.84 |
| Print Time (s) | 11.70 | 1.58 | 0.28 | 15.10 | 14.50 | 14.81 |
| Sol. Time (s) | 28452.5 | 3654.74 | 613.37 | 415.14 | 207.55 | 51.87 |
| Cost ($) | 0.94 | 0.97 | 0.97 | 0.19 | 0.19 | 0.19 |
| Energy (J) | 227619.90 | 233903.34 | 235535.59 | 13699.54 | 13698.26 | 13693.02 |
| AWS Instance | m5n.large | m5n.4xlarge | m5n.24xlarge | f1.2xlarge | f1.4xlarge | f1.16xlarge |

*Note:* Execution, initialization, printing and solution time (in seconds), cost (in USD) and energy (in joules) to solve 1,200 baseline economies using `Open MPI` CPU multi-core acceleration on Amazon M5N multi-core instances (with 1, 8, 48 physical cores, Columns 1-3) and using FPGA acceleration on Amazon F1 instances (connected to 1, 2, 8 FPGA devices, Columns 4-6).

Table A.5: CPU Performance by Grid Size

| Aggregate Capital, $N_M$ | 4 | | | 8 | | |
|---|---|---|---|---|---|---|
| Individual Capital, $N_k$ | *100* | *200* | *300* | *100* | *200* | *300* |
| Exec. Time (s) | 28464.55 | 51007.22 | 77061.15 | 71762.40 | 143718.80 | 229127.68 |
| Init. Time (s) | 0.36 | 0.38 | 0.39 | 0.37 | 0.40 | 0.41 |
| Print Time (s) | 11.70 | 12.72 | 14.94 | 14.38 | 15.94 | 18.38 |
| Sol. Time (s) | 28452.5 | 50994.12 | 77045.81 | 71747.64 | 143702.46 | 229108.89 |
| Cost ($) | 0.94 | 1.69 | 2.55 | 2.37 | 4.75 | 7.57 |
| Energy (J) | 227619.90 | 407952.96 | 616366.51 | 573981.11 | 1149619.67 | 1832871.10 |

*Note:* Execution, initialization, printing and solution time (in seconds), cost (in USD) and energy (in joules) to solve 1,200 baseline economies on a single core CPU (m5n.large) for different grid sizes (columns) on individual capital $N_k = \{100, 200, 300\}$ and aggregate capital $N_M = \{4, 8\}$.

Table A.6: Precision Accuracy Analysis

Panel A: ALM Coefficients

| | $\beta_1(a_b)$ | $\beta_2(a_b)$ | $\beta_1(a_g)$ | $\beta_2(a_g)$ |
|---|---|---|---|---|
| Floating-Point | 0.1460 | 0.9599 | 0.1554 | 0.9587 |
| Fixed Point | 0.1460 | 0.9599 | 0.1554 | 0.9587 |

Panel B: Policy Function, $k'$

| $\text{Mean}\left(\frac{|\text{Fixed}-Float|}{Float}\right)\%$ | 4.0e-10 | $\text{Max}\left(\frac{|\text{Fixed}-Float|}{Float}\right)\%$ | 2.6e-08 |
|---|---|---|---|

Panel C: Individual Capital Holdings Distribution, $T = 1,100$

| | Mean | Std | 0.25 | 0.5 | 0.75 |
|---|---|---|---|---|---|
| Floating-Point | 40.49 | 133.44 | 12.23 | 16.00 | 19.78 |
| Fixed Point | 40.49 | 133.44 | 12.23 | 16.00 | 19.78 |

| $\text{Mean}\left(\frac{|\text{Fixed}-Float|}{Float}\right)\%$ | 2.4e-09 | $\text{Max}\left(\frac{|\text{Fixed}-Float|}{Float}\right)\%$ | 3.0e-08 |
|---|---|---|---|

Panel D: Euler Equation Errors (EEE)

| | EEE | FPGA | CPU | $|\Delta_{\text{FPGA}-\text{CPU}}/\text{CPU}|\%$ |
|---|---|---|---|---|
| $N_k = 100$ | Mean (%) | 0.12 | 0.12 | 1.35e-07 |
| | Max (%) | 1.03 | 1.03 | 4.85e-07 |
| $N_k = 300$ | Mean (%) | 0.14 | 0.14 | 3.29e-07 |
| | Max (%) | 0.21 | 0.21 | 1.83e-07 |