# 20 Years of Static Dataflow

Oskar Mencer Founder Maxeler Technologies

Nov 2021

2001

#### **Computer Systems Laboratory Colloquium**

4:15PM, Wednesday, May 2nd, 2001 NEC Auditorium, Gates Computer Science Building B03

#### **Computing with FPGAs**

Oskar Mencer Lucent / Bell Labs, and Imperial College, London

#### About the talk:

Field-Programmable Gate Arrays (FPGAs) can outperform microprocessors on certain tasks by many orders of magnitude. The open research problems of computing with FPGAs are: (1) understanding the limitations of FPGAs when competing with microprocessors, and (2) providing a useful programming methodology.

First, I will show how FPGAs can be utilized to accelerate certain algorithms by up to three orders of magnitude. Examples for methods achieving the speedups are:

- a. exploring parallelism and pipelining on the bit-level,
- b. optimizing the encoding of data values (number representation), and/or
- c. reducing the required memory bandwidth by implementing data-structures and algorithms directly on the FPGA.

In addition, the speedup could be translated into savings in power consumption.

Second, I suggest a programming methodology for FPGAs based on Domain Specific Compilers. Domain specific compilers implement a divide-and-conquer, bottom-up approach to programming FPGAs. The vast space of possible architectures fragments into architecture families, which indirectly defines application domains. A domain specific compiler targets one architecture family and thus focuses on a single application domain. The StReAm compiler, under development at Bell Labs and Imperial College targets pipelined data-flow graphs mapped directly from object-oriented C++ to hardware. The goal is to provide a simple abstraction for programming FPGAs analogous to the abstraction of a microprocessor provided by the C programming language.

#### Euclids Elements, Representing a<sup>2</sup>+b<sup>2</sup>=c<sup>2</sup>

adverning in fight the start of any hilder in the second all guirding of any where a prove of a general ships and all all the stand the second should be again and all and you there will have a by to me the ort ALS TIM A glabalant the angle of the states among hereing

1 the land - operation and a second second Languer and allow up month for a march of Sylperial Chargen pipelaten

and the second realizing my after for manping minapping The state of the s the state of the second The interesting

and that the same 1 St more working PROPER TOTAL TOWNER TOWNER Lo Karman Inal - State gant - 14

Trang - apple to a sub why by the harden in open dies - im second his Adad when I wohyll to low yes - mp in a my - , by pay my arm is dy a here a good as in on you he up . fry musp - 1 a a & by due Himp an aporta haver haily

al and a marine and a film gagg hear in the hay hear hear he concupated of hashed inthe

ay 146 h apprairy my ire a vay ash back while a bear press between 5

of the as le age pair a province his is upon going has book ag is ago a h pilifiopra abra pilighted up. hal - 1 - pepperpe dumbi a signaly -ponte-jour ve asave ponter

an apa hily to my hish heave inghability of the ary him to dy for to go and more your to a for has to g jo high for

ב אינה פי עדור ב אועושור אואד ובריו אי

1 2 Page 2 to a Ke. Bat + Br wer real

man and indiand and the + time of the of fairly any hope agoon horas any of hapes a we share it in the war a gride by

a how in hora jo hay for hand is har going

and a share and a sound of the second to and and a to the share with reason of the house of the second of the seco and like at alimo say " Som bur, what the in a a new the live wan by a de new Down for the may uspen in a subman of Di wyor it there the

age have go ghian hush is has - as have all - 20 my - I a mile 14 line To Guard harry top an harry mpins harms your Ampian - Lye Limy roll . Laps. . hard and a surface strate The A when ye to land restriction by ac any - i li hanne year pro lab days

" a papel by a paper of a a ler the I per and per thank he have I pake

I also restate . del a adabt por a state By yoursple fax this when you tan land. the an iligination and his has ma man up of his derphis frought the sus to al wat also vist also - 22 mapat Toptala data of a poil wh Tomphy men far jou all here of to jo apagas de pluare out ma Talk a bay take and in the state man places fiel 2 & A bally making a to ale by to be for least of the and all phall about lash I are al. Taked what of shaderland al - illian while a - the large par - A ded above la abally was by the popular a the by information many.

amonghon grapy pe pairie ho ARTION CONTRACTOR PARTY SECOND

I min to I want of Song To work it was to a ,

Leveral galabrara like where. to amore the planop & have you histo is an id house por by The Ambahilger of the James marguer limepolities - a lipe of an a what y to be high a marrie o why the hope agains interit

winn the of aft group to ware.

Look with - toris - wine - tork war + A

ath property of mpart - his pa

The man from - Anna of the of the former of the second

Tray Forter in the man or other adjust at the the the the the Bapinen Filter and we have gail and " The The The Ared way he

-1-08 To Krist KE TOBALE the print 2 A ENTR a mosti will Grappy my house a company of my my do

hab 3 flor abob lash pilors are a Indian and Indian Indiany no mild prop - 1 Gp ay apair - h - - 6 p. Kangar Labra Justal a mp ( will be & mpan his af & gin. up of hay for - i bed mhoud above and it a generation of free a leader band de damband la ma lara

a the first and the second section of the second second second second second second second second second second

agoner + 19 - conthe second a more har Doug & Fam how Baine 2 malinhan of the back of a " parma pay any any open priver and haken you had good as a hand on it as an his of a more my own in any laws le for my he an Jahnah big lister & you hit shord Interity had optimately a the part is amouthing any fight poppinamo - the as shipy a pmile as post of an inder a second I have a final more that as

> amo I manual at a find what las AP 14 - reso departing and - 1 14A about ... etter tore hore may . I why you ad " I a should be be gue. ming him all y up to be y up my fy the give

Lay haben Juny pane. b as by sto up he for your and the as Jacks - part a believe baramen The inter a popular particita two-planag - for moments

hay with party & for him of he joh long ting salager a car au prophilo Vier plant Luray doe pators as by the foly have a for h & you do H agginty pravaparts a

prover for you a way jo have

reparator in to A . TO at Tomlere take tom 「いいうういち」 The Frend week mit 18 いっいいとうというですですです that many the lot 175 Part Matheren is me contra recourses in and いいか かんい おや しろちにや - and - wight way and + 2 " I WE TO THE TO BE . 121 & Ansie artery and the marker is the Busie meguine remend t's mint - williams to and a d'agener a'r Une they at the sea and a for the state of the stat 11 may interest and inches TTOM TONE L'E BALLY IN an grand with and and + Talmer aller und But any Brown agen The awar ward

25

matzeliterter

a conversion of 30 a sau Tumar 25 a arek werden

タんだ ションカーション

# 1971



The Intel 4004 microprocessor, which was introduced in 1971. The 4004 contained 2300 transistors and performed 60,000 calculations per second. Courtesy: Intel.



l It

AT A

#### Wires getting much bigger in size than transistors



# **2003: The Maxeler Static Dataflow Model**

let's create a computing structure to fit the problem

- Data moves continuously (flow) and drive computation
- Compute in Space arrange operations in 2D
- Find optimal solution for any *specific* flow problem
  - ◆ No wasted silicon maximum performance density
  - No wasted clock cycles data rate = clock rate
  - Predictable throughput & latency, MIN ENERGY for moving data





### a different way to compute?



### a warning from history...I did not listen...

LECTURE 45

26 AUGUST 1946

A PARALLEL CHANNEL COMPUTING MACHINE

Lecture by J. F. Eckert, Jr. Electronic Control Company

Again I wish to reiterate the point that all the arguments for parallel operation are only valid provided one applies them to the steps which the built in or wired in programming of the machine operates. Any steps which are programmed by the operator, who sets up the machine, should be set up only in a serial fashion. It has been shown over and over again that any departure from this procedure results in a system which is much too complicated to use.

- J. P, Eckert, Jr (Co-Inventor of ENIAC)

Credit: Prof. Paul H.J. Kelly

MAXELER

See also http://www.digital60.org/birth/themooreschool/lectures.html#45

Stanford EE382A



### John von Neumann, 1946:

"We are forced to recognize the possibility of constructing a *hierarchy of memories*, each of which has greater capacity than the preceding, but which is less quickly accessible."

So, clearly what matters is the location of data!!



#### As a result of the von Neumann hierarchy:

#### ☑ latency.txt Assembly Instruction: LD A, (B) at 2 GHz

1	Latency Comparison Numbers						
2							
3	L1 cache reference	0	.5 ns				
4	Branch mispredict	5	ns				
5	L2 cache reference	7	ns				
6	Mutex lock/unlock	25	ns				
7	Main memory reference	100	ns				
8	Compress 1K bytes with Zippy	3,000	ns	3	us		
9	Send 1K bytes over 1 Gbps network	10,000	ns	10	us		
10	Read 4K randomly from SSD*	150,000	ns	150	us		
11	Read 1 MB sequentially from memory	250,000	ns	250	us		
12	Round trip within same datacenter	500,000	ns	500	us		
13	Read 1 MB sequentially from SSD*	1,000,000	ns	1,000	us	1	ms
14	Disk seek	10,000,000	ns	10,000	us	10	ms
15	Read 1 MB sequentially from disk	20,000,000	ns	20,000	us	20	ms
16	Send packet CA->Netherlands->CA	150,000,000	ns	150,000	us	150	ms
17	SQL Database Transactions		> 150	0,000 us	>	150r	ns
R						10	

#### **PYTHON vs SQL: Real Customer Project**



Running Python: Duration: 0.952739 seconds

speedup(p, s)

Speedup: 119.9x





#### **Kolmogorov Complexity, 1965**

**Definition** (Kolmogorov): "If a description of *string s*, *d*(*s*), is of minimal length, [...] it is called a **minimal description** of *s*.



the length of d(s), [...] is the **Kolmogorov complexity** of *s*, written K(s), where  $K(s) = |d(s)|^{n}$ 

Of course K(s) depends heavily on the *Language L* used to describe actions in K (e.g., Java, Esperanto, an Executable file, etc).

Kolmogorov, A.N. (1965). <u>"Three Approaches to the Quantitative Definition of Information"</u>. Problems Inform. Transmission 1 (1): 1–7.



First Large Scale Static Dataflow 4,866 ALUs for a time step solving the Acoustic Wave Equation



# Impossible? or merely hard?





# MaxJ: A Dataflow Programming Model

Syntax based on Java, and Semantics for static dataflow



Making DATAFLOW programming fun,

"Easy, Desirable, and Affordable" [Terry Leahy, former CEO of TESCO]



# Ludwig Wittgenstein

Born: Vienna, Austria 1889 Died: Cambridge, England, 1951

The limits of my language mean the limits of my world

Dataflow Corollary: The limits of my programming language mean the limits of what I can optimize...



# A Dataflow Kernel Every line of code corresponds to a resource



### Dataflow can be annotated back into code

#### every line of dataflow code takes a certain space

BRAMs DSPs : MyKernel.java LUTs FFs 727 871 1.0 2 : resources used by this file 0.10% : % of available 0.24% 0.15% 0.09% 71.41% 61.82% 100.00% 100.00% : % of total used 94.29% 97.21% 100.00% 100.00% : % of user resources : public class MyKernel extends Kernel { public MyKernel (KernelParameters parameters) { super(parameters); DFEVar p = io.input("p", dfeFloat(8,24)); 31 0.0 0: 1 2 9 0.0 0: DFEVar q = io.input("q", dfeUInt(8)); DFEVar offset = io.scalarInput("offset", dfeUInt(8)); : 8 8 0.0 0: DFEVar addr = offset + q; 18 1.0 40 0: DFEVar v = mem.romMapped("table", addr, dfeFloat(8,24), 256); : 2 : 139 145 0.0 p = p \* p;0.0 401 541 0: p = p + v;io.output("r", p, dfeFloat(8,24)); : }

# Putting it all together: A Dataflow System Architecture





### Kalman Filter as a State Machine in MaxJ

class TrackFitKernel extends Kernel{

protected TrackFitKernel(KernelParameters p) {

$$\begin{aligned} \sup_{k=1}^{k-1} &= \mathbf{F}_{k-1} x_{k-1} & \text{Stub stubIn} = \text{Stub.input(this, "stubIn");} \\ \sum_{k=1}^{k-1} &= \mathbf{F}_{k-1} \mathbf{C}_{k-1} \mathbf{F}_{k-1}^{T} + \mathbf{Q}_{k-1} & \text{State kStateUp} = \text{KF.update();} \\ \sum_{k=1}^{k-1} &= m_k - \mathbf{H}_k x_k^{k-1} & \text{State kStateUp} = \text{KF.update();} \\ \sum_{k=1}^{k-1} &= m_k - \mathbf{H}_k x_k^{k-1} & \text{State kStateUp} = \text{KF.update();} \\ \sum_{k=1}^{k-1} &= \mathbf{V}_k + \mathbf{H}_k \mathbf{C}_k^{k-1} \mathbf{H}_k^T & \text{State kStateUp} = \text{KF.update();} \\ \sum_{k=1}^{k-1} &= \mathbf{V}_k + \mathbf{H}_k \mathbf{C}_k^{k-1} \mathbf{H}_k^T & \text{State kStateUp} = \text{KF.update();} \\ \sum_{k=1}^{k-1} &= \mathbf{V}_k + \mathbf{H}_k \mathbf{C}_k^{k-1} \mathbf{H}_k^T & \text{State kStateUp} = \text{KF.update();} \\ \sum_{k=1}^{k-1} &= \mathbf{V}_k + \mathbf{H}_k \mathbf{C}_k^{k-1} \mathbf{H}_k^T & \text{State kStateUp} = \text{KF.update();} \\ \sum_{k=1}^{k-1} &= \mathbf{V}_k + \mathbf{H}_k \mathbf{C}_k^{k-1} \mathbf{H}_k^T & \text{State kStateUp} = \text{KF.update();} \\ \sum_{k=1}^{k-1} &= \mathbf{V}_k + \mathbf{H}_k \mathbf{C}_k^{k-1} \mathbf{H}_k^T & \text{State kStateUp} = \text{KF.update();} \\ \sum_{k=1}^{k-1} &= \mathbf{K}_k^{k-1} \mathbf{H}_k^{k-1} \mathbf{H}_k^{k-1} & \text{State kStateUp} = \text{KF.update();} \\ \sum_{k=1}^{k-1} &= \mathbf{K}_k^{k-1} \mathbf{H}_k^{k-1} \mathbf{H}_k^{k-1} & \text{State kStateUp} = \text{state.x();} \\ \sum_{k=1}^{k-1} &= \mathbf{K}_k^{k-1} \mathbf{H}_k^{k-1} \mathbf{H}_k^{k-1} & \text{Matrix pxx} = \text{state.pxx();} \\ \sum_{k=1}^{k-1} &= \mathbf{K}_k^{k-1} \mathbf{H}_k^{k-1} \mathbf{H}_k^{k-1} & \text{Matrix pxd} = \text{pxx} \times \mathbf{H}.\text{transpose();} \\ \sum_{k=1}^{k-1} &= \mathbf{K}_k^{k-1} \mathbf{H}_k^{k-1} \mathbf{H}_k^{k-1} & \text{Ketor residual} = \mathbf{d} - \mathbf{h}_k; \\ \sum_{k=1}^{k-1} &= \mathbf{K}_k^{k-1} \mathbf{H}_k^{k-1} \mathbf{H}_k^{k-1} & \text{Ketor xUp} = \mathbf{x} + \mathbf{K} \times \text{residual}; \\ \sum_{k=1}^{k-1} &= \mathbf{K}_k^{k-1} \mathbf{H}_k^{k-1} \mathbf{H}_k^{k-1} & \text{Ketor xUp} = \mathbf{x} + \mathbf{K} \times \text{residual}; \\ \sum_{k=1}^{k-1} &= \mathbf{K}_k^{k-1} \mathbf{H}_k^{k-1} & \text{Ketor xUp} = \mathbf{K}_k^{k-1} \mathbf{K}_k^{k-1} & \text{Ketor xUp} = \mathbf{K}_k^{k-1} \mathbf{K}_k^{k-1} & \text{Ketor xUp} = \mathbf{K}_k^{k-1} \mathbf{K}_k^{k-1} & \text{Ketor xUp} = \mathbf{K}_k^{k-1} & \text{Ketor xUp} & \text$$





# **Parallel Loop with Dependency**



for i=1 to N: // sequential loop
for j=1 to 4: // data parallel loop
out[j] = out[j]+stream\_in[i];

Of course j could be a lot larger, but we do **4 at a time** here since we assume 4 stages in a

MAXELER

DFEParLoop lp2 = new DFEParLoop(this, "lp2"); DFEVar in = io.input("in", dfeFloat(8, 24), lp2.ndone); lp2.set\_input(dfeFloat(8,24), 0.0);

DFEVar result = in + lp2.feedback; lp2.set output(result);

io.output("result", result, dfeFloat(8, 24), lp2.done);



# Prof Mike Flynn, Stanford

https://www.youtube.com/watch?v=ybnOul9jNgE from 4'50"

#### Slotnick's law (of effort) from the great debate with Gene Amdahl, 1967

"The parallel approach to computing does require that some original thinking be done about numerical analysis and data management in order to secure efficient use.

In an environment which has represented the absence of the need to think as the highest virtue this is a decided disadvantage."

-Daniel Slotnick

# **Maxeler Dataflow Engines (DFEs)**



High Density DFEs Intel Xeon CPU cores and up to 6 DFEs with 576GB of RAM





**The Dataflow Appliance** Dense compute with 8 DFEs, 768GB of RAM and dynamic allocation of DFEs to CPU servers with zero-copy RDMA access



**The Low Latency Appliance** Intel Xeon CPUs and 1-2 DFEs with direct links to up to six 10Gbit Ethernet connections





#### MaxWorkstation Desktop dataflow development system

MaxRack

10, 20 or 40 node rack systems integrating compute, networking & storage

MaxCloud Hosted, on-demand, scalable accelerated compute **Dataflow Engines** 48GB DDR3, high-speed connectivity and dense configurable logic



# 2013 ... getting another 10-20x of speedup if you optimize Flow of data with Maxeler hardware

# Simulating the Atmosphere via the Shallow Water Equation

[L. Gan, H. Fu, W. Luk, C. Yang, W. Xue, X. Huang, Y. Zhang, and G. Yang, Accelerating solvers for global atmospheric equations through mixed-precision data flow engine, FPL Conference 2013]

Platform	Performance	<u>Speedup</u>	Efficiency	<u>Energy</u> Improvement
6-core CPU	4.66K	1	20.71	1
Tianhe-1A node	110.38K	23x	306.6	14.8x
MaxWorkstation	468.1K	100x	2.52K	121.6x 9x
Maxeler MPC-X	1.54M	330x	ЗK	144.9x

Deep

Shallow

Navier-Stokes

Euler

Boussinesa

Hydrostatic

Shallow water

# How is it possible to beat the worlds fastest computer by 14x on speed and 9x on energy

Paraphrased: In theory, computing units can be constructed which use no energy.

Energy is only needed when information is lost.

Reordering of **information** does not require energy from a pure physics perspective.

Of course, moving information takes Energy...

Dataflow minimizes Data Movement!





#### 2015: First Maxeler Dataflow Supercomputer installed at UK Government Laboratory



#### 20-50x increased compute capability per cubic-foot of data center space



#### 2017: Maxeler Dataflow Machine at Jülich in Germany



Technologies

#### How do we migrate applications to **Static Dataflow**

#### Start with a Maxeler Loop Flow Graph





#### **Convert the Loop Flow Graph into an Architecture**



#### **Example: Quantum Chromodynamics**





#### **Numerical Analysis: Maxeler Value Profiling**

We had to change how we implement (represent) numbers in computers, and manage numerics





Changing the way we represent numbers in computers



# Aerial View of a Neural Network on FPGA



and the

MAXELER

#### **Sparse Matrix Dataflow – can't be done?**



# **Think Big**





IIIt

### 2020: Inspired over 1,000 publication

≡	Google Scholar	Maxeler	(
٠	Articles	About 1,270 results (0.03 sec)	
	Any time Since 2020 Since 2019 Since 2016 Custom range	Sorting networks on Maxeler dataflow supercomputing systems <u>A Kos</u> , V Ranković, <u>S Tomažič</u> - Advances in computers, 2015 - Elsevier The primary contribution of this study is the implementation and evaluation of network sorting algorithms on a <b>Maxeler</b> dataflow computer. Sorting is extensively used in numerous applications. We discuss sequential, parallel, and network sorting algorithms. The major part $\Rightarrow 99$ Cited by 29 Related articles	



MAXEL

### 2020: Static Dataflow inspiring modern Chips

The Groq Architecture, see: Think Fast, A Tensor Streaming Processor for accelerating Deep Learning Networks, ISCA 2020.



Fig. 1. Conventional 2D mesh of cores (a) reorganized into a functionally sliced arrangement of tiles (b).

#### "The hallmark of [Data]flow is a feeling of spontaneous joy"



MAXELER