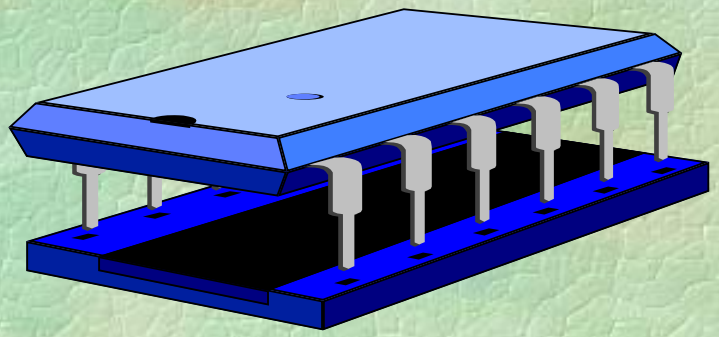
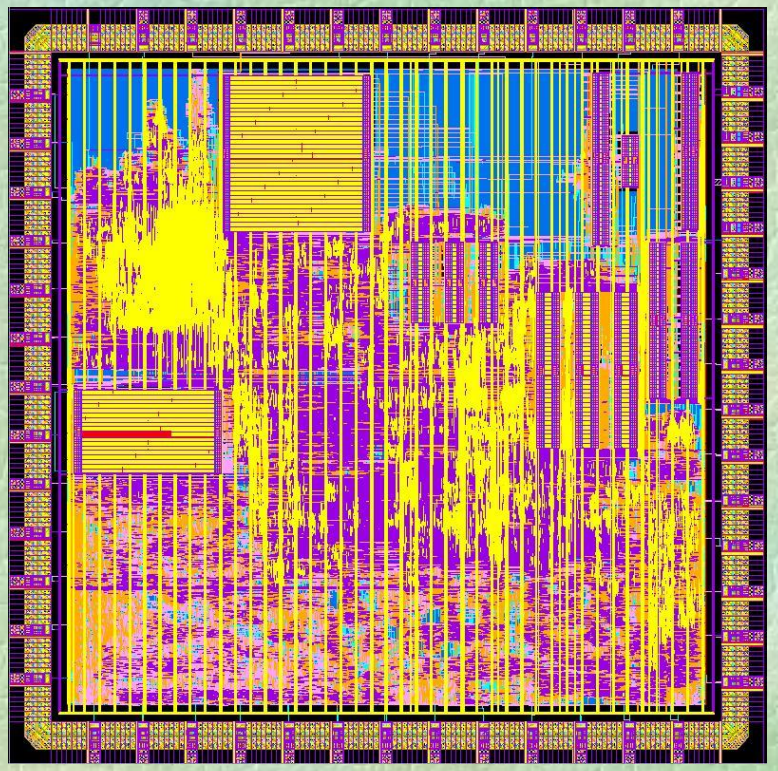
הפקולטה להנדסת חשמל
המעבדה ל- VLSI
מרכז מחקר למערכות VLSI





V - VERY

L - LARGE

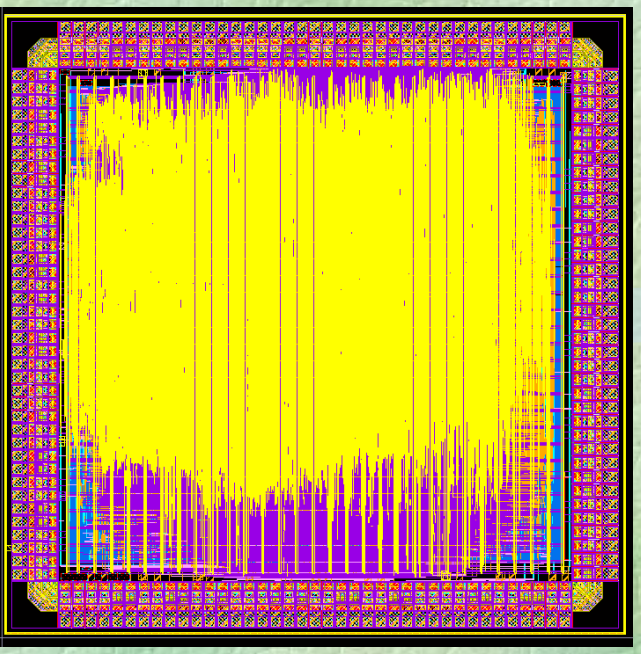
S - SCALE

I - INTEGRATION

מאות מיליוני טרנזיסטורים על שטח קטן

שוק של כ- 300 מיליארד דולר בשנה

פרסומות ☺



- **תחרות פרויקט מצטיין פקולטי**
  - מקום ראשון 2008, 2010, 2011, 2014 ו- 2016
  - מקום שני 2012 ו- 2013!
  - מקום שלישי 2015!
- **ייצור שבבים**
  - מחקר
  - הוראה : שבב שפותר בעיית "הסוכן הנוסע"
  - הוראה : שבב המממש רשת נוירונים

המעבדה ל- VLSI

צוות המעבדה :

• גואל סמואל – מהנדס המעבדה
• חדר : 711 טל : 4668, goel@ee.technion.ac.il

• אמיר בר – מהנדס מעבדה
• חדר 714, טל : 4671, baer@ee.technion.ac.il

• שרון בר-לב שפי - מהנדסת מחקר
• חדר 711, טל 4668, sharonb@tx.technion.ac.il

חדרי המעבדה ל- VLSI

- 711 - חוות UNIX
- 714 - חוות PC + מעבדה בדיקות
- 715 - חוות UNIX + PC
- 718 - מעבדה Memristors
- 719 – חדר דיונים

הנחיות כלליות

- המעבדה פתוחה 24 שעות ביממה – 7 ימים בשבוע
- אין לעבוד ביחידות במעבדה (בטיחות)
- ניתן לעבוד "מרחוק" – (לפנות לאמיר)
- אין לכבות או לבצע reset למחשבים במעבדה
- אין לנתק את התקשורת של המחשבים במעבדה (גם לא באופן זמני)
- אין לאכול או לשתות במעבדה
- לשמור על ציוד המעבדה

ציוד במעבדה ל- VLSI

- 20 Linux Workstations / Servers
- 4 Sun/Solaris Workstations
- 10 Windows Workstations
- 16903 Agilent DA+PG
- Analog Test Equipment
  - Digital Oscilloscope - 1Ghz
  - Arbitrary waveform generator - 125 Mhz
  - Quad power supply
  - Metallurgical Microscope

VLSI SYSTEMS RESEARCH CENTER

VLSI LABORATORY

DEPARTMENT OF ELECTRICAL ENGINEERING DEPARTMENT OF COMPUTER SCIENCE

Typical VLSI Design Flow

IDEA

ARCHITECTURAL DESIGN

CUSTOM BLOCKS

SUB-BLOCK SCHEMATIC

TRANSISTOR LEVEL-SIMULATION

LAYOUT

EXTRACTION

DRC and LVS

POST-LAYOUT SIMULATION

HDL BLOCKS

LOGIC SIMULATION

LOGIC SYNTHESIS TECHNOLOGY MAPPING

TIMING VERIFICATION

AUTOMATIC PLACE & ROUTE

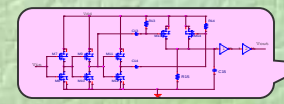
POST-LAYOUT SIMULATION/TIMING

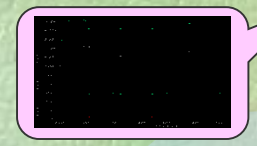
TOP-LEVEL PLACE & ROUTE

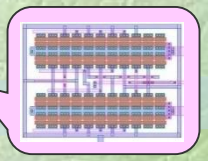
TOP-LEVEL VERIFICATION

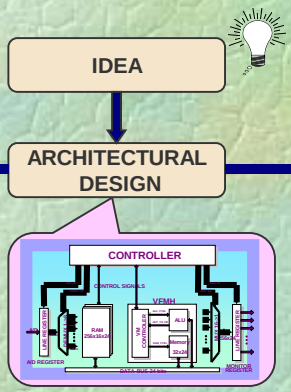
PROTOTYPING

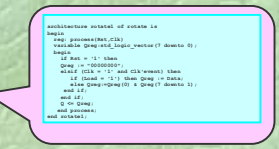
TESTING

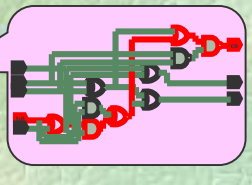


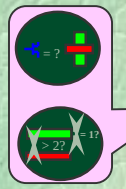


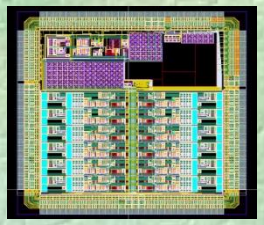


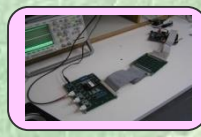


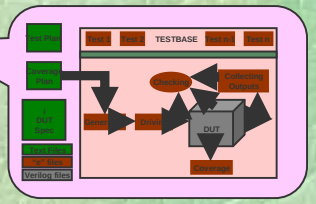


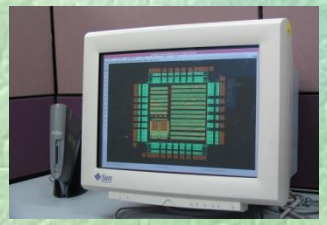












תוכנות תיב"מ - ספרתי

1. Design Entry : SystemVerilog, VHDL, Bluespec SV
   - Language Sensitive Editors
2. Simulation / Verification
   - Ncsim, VCS
   - Specman
3. RTL Synthesis
   - Design Compiler, Genus
4. Timing Verification
   - Design Compiler, Primetime
5. Physical Design
   - Innovus, ICC

- תוכנות של חברות Cadence ו- Synopsys
- מומלץ לעבוד על תוכנות המעבדה
- ניתן לעבוד מרחוק

תוכנות תיב"מ - אנלוגי

1. **Design Entry : Transistor Level Schematics**
   - **Composer**
2. **Circuit Simulation**
   - **cdsSpice, Spectre/S, ADS**
3. **Handcrafted Layout**
   - **Virtuoso, Layout GXL**
4. **Layout Verification**
   - **Assura, PVS, Calibre**

- תוכנות של חברות Synopsys, Cadence, Mentor, Keysight
- מומלץ לעבור על תוכנות המעבדה
- ניתן לעבוד מרחוק

סימולציות אלקטרומגנטיות

- HFSS

סימולציות מכניות וחום

- COMSOL Multiphysics -> HFSS

Optical Simulations

- Phoenix Software Tools

תהליכי יצור

- Tower CMOS 0.18u 8LM
- Tower CMOS 0.13u 6LM
- ST Microelectronics 0.13u
- TSMC 65nm CMOS
- Global Foundries CMOS RF 40nm and 28nm

דרישות כלליות

- הוראות בטיחות
- דו"ח איפיון
- מסמך ספציפיקציות
- מצגת אמצע סמסטר – אחרי 8 שבועות
- מצגת סופית – בסוף הפרויקט
- דו"ח סופי
- קשר שבועי עם המנחה
- השקעה : כ- 12 שעות בשבוע
- הצגת התוצאות סופיות למנחה
- סיום פרויקט – סוף סמסטר

השיטה ההנדסית

1. הבן את הבעיה
2. אילו אמצעים דרושים למציאת הפתרון ?
3. בחן את האלטרנטיבות השונות לפתרון
4. בחר את הפתרון הטוב ביותר ופתח דרך שיטתית לפתרון
5. בצע את התוכנית
6. נתח את הפתרון – תקן חלקים שלא עובדים
7. תעד את התוצאות

חלוקת האחריות

- מנחה
  - אחריות מקצועית/אקדמית על הפרויקט
- צוות המעבדה
  - אחריות על הציוד, תוכנות, תשתית

בעיות

- יש לפנות מיד לגואל עם כל בעיה הקשורה לביצוע הפרויקט !
- לא לחכות עד סוף הסמסטר!

אתר המעבדה

- manuals
- תיאור מקוצר ל- VHDL
- SystemVerilog
- סינתזה
- layout
- ועוד ....

- כיצד להכין מצגת
- כיצד לכתוב דו"ח
- useful links

www.ee.technion.ac.il/vlsi

פרויקטים ב- VHDL או (System)Verilog

- יש לסיים את התכנון לפני תחילת המימוש
- חשוב לכתוב בסגנון נכון
- המימוש צריך להיות סינתזבילי
  - תיאור מקוצר ל- VHDL
  - מבוא ל- SystemVerilog
- מומלץ להתייעץ עם גואל לפני תחילת המימוש
- מומלץ לבדוק "סינתזביליות" כבר מהיחידה הראשונה
- סינתזה ו- layout

במשאבי המעבדה

- חשבונות על מחשבי Linux
- קידוד כרטיסים
- יש למלא טפסים אצל אמיר בשבועיים הראשונים

- **החשבונות אישיים ואין לאפשר לאחרים להשתמש בהם**

השימוש במשאבי המעבדה :

- חשבונות מחשב
- מדפסות

לצרכי הפרויקט בלבד !

ViLoCoN – Video Lossless Compression for Network on Chip

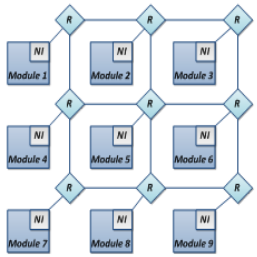
Shani Rehana and Or Turgeman - Supervised by Ran Manevich

TECHNION – ISRAEL INSTITUTE OF TECHNOLOGY , EE Department – VLSI Laboratory

Runners Up Kasher EE Project Competition 2013

NoC with Lossless Compression

- ***Network On Chip (NoC)*** - Network based communication
- **Uses:** Routers and packets
- **Advantages:** Enhanced connectivity, throughput and bandwidth
- Compress/decompress inside the Network Interface (NI)
- **ViLoCoN** - A lossless compression CODEC



NoC Architecture

Project Goals

Design and Implement a ViLoCoN Codec

(Video Lossless Compression for NoCs)

Encoder and Decoder Requirements:

- **Area:** Ultra-lightweight – less than 2[Kgates]
- **Latency:** Approximately 10 clock cycles
- **Frequency:** Real time performance - 1[GHz] in 65[nm] technology.
- **RTL language:** "System Verilog"

ViLoCoN Algorithm

- Variable length differences sequences (DS) encoded with codewords of fixed length

DCM: **D**ifferences **C**odes **M**ap Table: Encodes the most common differences sequences

N: Maximal length of differences sequences in DCM

C: Length of codeword

Input - Raw pixels sequence – 64 bits

45	46	47	48	49	61	61	61

Differences sequence (DS)

45	1	1	1	1	12	0	0

ViLoCoN output – 25 bits

$(45)_8$	1	$(5)_3$	0	$(12)_8$	1	$(0)_3$
raw	DB	code	DB	raw	DB	code

DCM – 3 bits codewords

DS	Codeword
(0,0)	0
(0,1)	1
(0,-1)	2
(1,1)	3
(1,1,1)	4
(1,1,1,1)	5
(1,1,1,2)	6
(1,1,1,-2)	7

Selection of ViLoCoN parameters

- Finding optimal N, C and DCM parameters:
  - Selection of 10 BW benchmarks images in 2 sizes: "Full HD" and "VGA"
  - Selection of a preliminary range for N and C: $2 \le N \le 12$, $7 \le C \le 14$
  - Identification of the 2^C most common difference sequences and creation of the "Optimal DCM table"
  - Image compression for every N and C in given range and calculation of compression ratio
- ***N=4, C=10:*** Provides best ***tradeoff*** between compression ratio and implementation area
- Optimal DCM table ⇨ requires LUT implementation ⇨ Large area
- ***Solution:*** Approximate the "optimal DCM" by most common continuous sets of sequences ⇨ Store only the boundaries of the sequences
- The approximated DCM table:

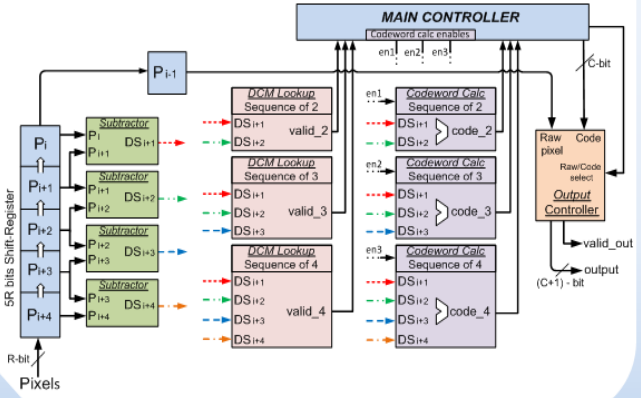
Sequence length	2	3	4
Upper boundary	(10 , 10)	(3 , 3 , 3)	(2, 2, 2, 2)
Lower boundary	(-9 , -10)	(-3 , -3 , -3)	(-1 , -1 , -1 , -1)

- ***Cost:*** Codewords need to be ***calculated*** from the boundaries.

  Formula for calculating codeword from a difference sequence of 2:

  $Codeword_2 = 21 \cdot (DS_{i+1} - (-9)) + (DS_{i+2} - (-10))$
- Similar formulas exists also for all codewords

Encoder Architecture



Decoder Architectural Design

- The difference sequence is calculated from the codeword :

  DS2 - (d1,d0) is calculated as follows:

$$d1 = \frac{codeword}{21} + (-9) \quad , \quad d0 = codeword \% 21 + (-10)$$

- Efficient division and modulo calculation. Example (divide by 7) :

$$quotient = floor\left(\frac{x + x \cdot 2^3 + x \cdot 2^6 + x \cdot 2^9}{2^{12}}\right)$$

Results and Comparison

- ***Area:*** Encoder: 1.8[KGEs], Decoder: 3.6[KGEs]. 71% and 84% smaller respectively, compared to other published lightweight encoders and decoders
- ***Compression ratio:*** Average compression ratio: 2.37, equivalent to a reduction of 57.8% data.
- ***Operation frequency:*** Encoder: 990/893[MHz] (2 options), Decoder: 905[MHz].
- ***Dynamic power:*** Encoder: 1.94[mW], Decoder: 4.2[mW].

Comparison with published designs:

Algorithm		ViLoCoN	DAP + Huffman	Sig. Bit Trunc.	Dict. Based	DWT+ SPIHT	FELICS+ CDP	HPE	Hwang's
Compression ratio (Y)		2.175	NA	2.14	NA	NA	2.35	NA	NA
Compression ratio (YUV)		2.37 YCbCr	1.81	2.52	2.18 YCbCr	2.00	2.31 RGB	3.09	2.53 YCbCr
Area	Logic (GEs)	5.4K	7K	36.1K	23.9K	27K	>12.97K	15.7K	>35K
	Memory (Byte)	0	0	0	5.4K	1.28K	1.9K	0.5K	3.7K
CEff (Y)		0.4	NA	0.059	NA	NA	<0.054	NA	NA
CEff(YUV)		0.45	0.258	0.069	0.019	0.042	<0.053	0.13	<0.026

- Codec efficiency (CEff) defined as:

$$CEff = \frac{Compression\ Ratio}{Codec\ Logic\ [K\ NAND\ gates]}$$

Original Contributions

- Development, design and implementation of a completely new encoder and decoder for NoC video applications
- Comprehensive analysis and engineering trade-offs to determine the optimal parameters for the ViLoCoN algorithm – N, C, and DCM table
- Implementation of the most lightweight encoder and decoder
  - Smart lookup and codeword calculations
  - Output stream controller – fast output ordering using small buffer
  - Division and modulo in an effective/economical technique

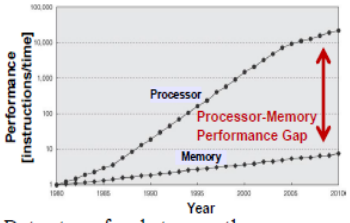
Controller for Memristor Based Logic

Rotem Ben-Hur, Hani Bezalel - Supervised by Dr. Shahar Kvatinsky

TECHNION – ISRAEL INSTITUTE OF TECHNOLOGY , EE Department – VLSI Laboratory

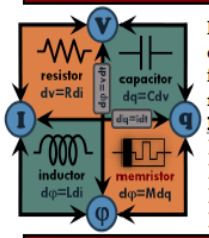
Memory Wall

The performance of current computer systems is limited by the speed of the memory.



Data transfer between the memory and the processor is time consuming and wasteful in energy.

Memristor

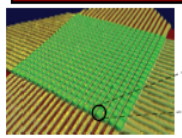


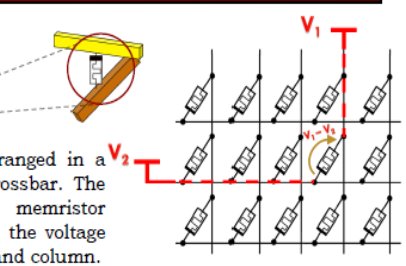
In 1971, Prof. Leon Chua predicted the existence of the memristor, a fourth fundamental element that defines the relation between magnetic flux and charge.

Memristor Properties:

- Modeled as a varying resistor
- Information is stored as resistance
- Nonvolatile, thus low power consumption
- Dense and fast at least as DRAM

Memristor Based Crossbar Memory

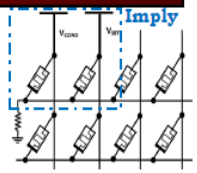




Memristors are arranged in a 2D array called crossbar. The resistance of a memristor varies according to the voltage applied on its row and column.

Complete Logic Structure in a Crossbar

It is possible to perform two basic logic operations - Imply and False - within a crossbar memory. These two operations create a complete logic structure, thus enabling **in memory processing**.



Imply Truth Table

Case	p	q	p→q
1	0	0	1
2	0	1	1
3	1	0	0
4	1	1	1

In-Memory **Imply** can be implemented by applying voltages on two memristors in the same row/column. **False** is simply done by writing zeros.

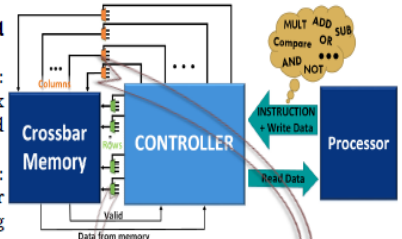
A Novel Architecture

Project Goal: Design and Implement a Novel Computer Architecture for In Memory Processing

To achieve this goal, a complex memory controller was designed and implemented.

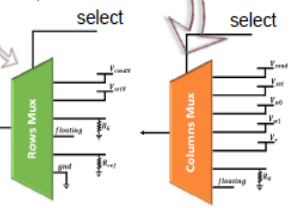
Unique properties (compared to traditional systems):

- Controller-Processor Interface: able to manage complex instructions, as well as standard read and write commands.
- Controller-Memory Interface: mux-based hardware for coordination between analog memory and digital controller.
- Complex Memory Controller: performs standard read/write operations, as well as processing tasks inside the memory.



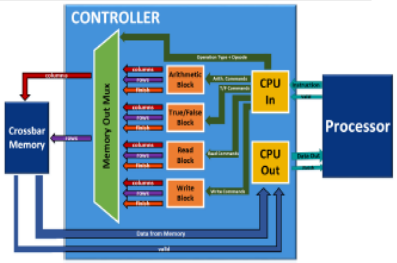
Analog muxes for controller-memory IF:

- Inputs are analog voltages.
- 'Select' input is a digital control signal, which arrives from the controller.

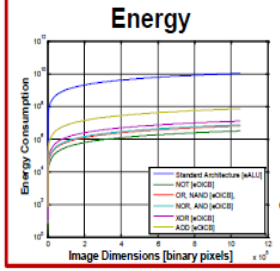


Our Memory Controller

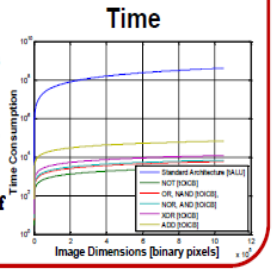
The controller receives commands from the processor and interprets them by dividing them into hundreds of micro commands. It then sends the appropriate control signals to all the relevant rows and columns in the memory.



Comparison of Energy & Time Consumption



The suggested architecture has the potential to dramatically reduce time and energy required to perform different kinds of processing.



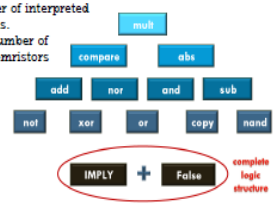
Complex in Memory Algorithms

All complex operations in the memory are based on two simple operations - Imply + False.

Example: In-memory multiplication algorithm:

```
N = length(A) = length(B), i=0
WHILE (i<N) {
    //Copy a under B:
    (a_i -> 0) -> 0          1+N cycles
    // a AND B:              1+N operations
    (a_i -> (B -> 0)) -> 0   3 cycles
    i++                      3N operations
}
//A MULT B:                  14+3N cycles
a_0 AND B + a_1 AND B0 +...+ a_N-1 AND B0...0   4+22N operations
                                                N-1 zeroes
```

Cycles = number of interpreted micro-commands.
Operations = number of participating memristors



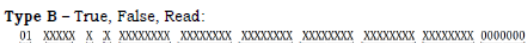
Total Complexity
Time – $4N^2+15N-14$ cycles
Energy – $26N^2-17N-4$ operations

Our Instruction Set

Designed a completely new, custom-made, instruction set:

Type A – Arithmetic/Logic:

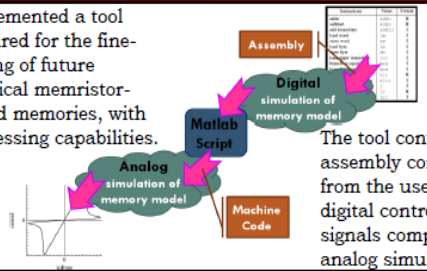
Type B – True, False, Read:



Type C – Write (begin) :

"Compiler" for in Memory Processing Architecture

Implemented a tool required for the fine-tuning of future physical memristor-based memories, with processing capabilities.



The tool converts assembly commands from the user, through digital control signals to signals compatible to analog simulation tools.

Our Contributions

- ✓ Novel approach for computer architecture design – towards mixed logic and memory.
- ✓ Has the potential to significantly improve power and time consumption compared to conventional systems.
- ✓ Main design contributions:
  - ✓ Memory controller that performs logic operations by conducting hundreds of memory micro instructions
  - ✓ Completely new instruction set
  - ✓ "Compiler" for the new architecture

Optical Permeability Bragg Accelerator

Elron Goldemberg & Almog Zilka, supervised by Adi Hanuka & Prof. Levi Schachter

1. Motivation- going optical

$\lambda = 10cm$

$Gradient \propto \frac{\sqrt{power}}{\lambda_L}$

$\lambda = 1\mu m$

Conventional accelerator

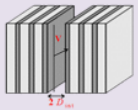
- Metallic device
- Based RF technology

Bragg accelerator

- Dielectric/permeability structure
- Based on optical frequency

Challenges

- Accelerating the electrons to relativistic speeds.
- Sustaining the high accelerating fields.
- Confining the electromagnetic energy.

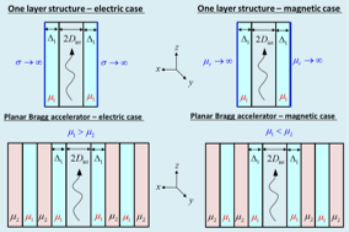


Our goals

- Designing of an optical planar Bragg accelerator consisting of **permeability** layers.
- Analyzing the acceleration parameters: $Z_{int}, V_{en}, \eta_{max}$
- Comparing to the dielectric Bragg accelerator.

2. Formulation

The proposed structures



Design procedure

- The Bragg structure consists of periodic permeability layers surrounding a vacuum core.
- Confinement is achieved by:
  - Adjusting the first layer width
  - Determining the rest of the layers width to be: $\frac{\lambda}{4\sqrt{\mu_r - 1}}$

Work flow

Finding the fields in the device

↓

Applying the boundary conditions

↓

Finding the first layer width

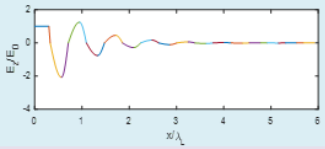
↓

Calculating the acceleration parameters

The electromagnetic fields

In the vacuum layer:	In the permeability layer:
$E_z = E_0 e^{-j\frac{\omega}{c}z}$	$E_z = (Ae^{-\beta_x x} + Be^{\beta_x x})e^{-j\frac{\omega}{c}z}$
$E_x = j\frac{\omega}{c} x E_0 e^{-j\frac{\omega}{c}z}$	$E_x = -\frac{(Ae^{-\beta_x x} - Be^{\beta_x x})e^{-j\frac{\omega}{c}z}}{\sqrt{\mu_r - 1}}$
$H_y = \frac{j}{\eta_0}\frac{\omega}{c} x E_0 e^{-j\frac{\omega}{c}z}$	$H_y = -\frac{(Ae^{-\beta_x x} - Be^{\beta_x x})e^{-j\frac{\omega}{c}z}}{\eta_0\sqrt{\mu_r - 1}}$

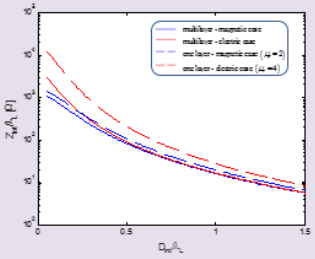
The tangential electric field along the structure



3. Results

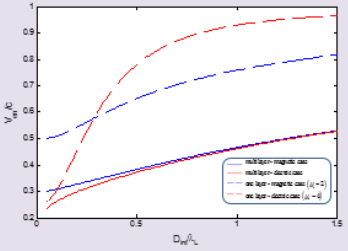
Interaction Impedance

$Z_{int} \triangleq \frac{|E_0|^2 \lambda^2}{P}$



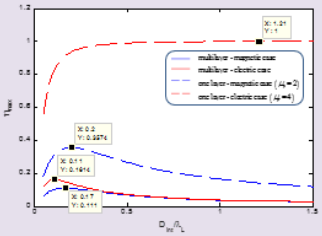
Energy Velocity

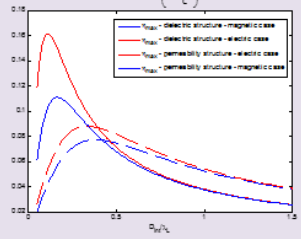
$V_{en} = \frac{P}{W_{EM}}$



Maximum Acceleration Efficiency

$\eta_{max} = \frac{Z_{int} V_{en}}{4\beta_L \Gamma\left(1 - \frac{V_{en}}{c}\right)\kappa}$ [1]





4. Conclusions

- The electric case achieves better performance among all structures.
- The permeability structure is better than the dielectric one, in respect to: Z_{int}, η_{max}

Reference

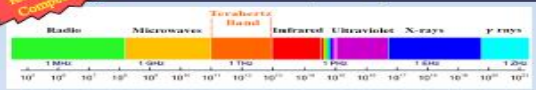
[1] K. Bane and G. Stupakov, Phys. Rev. Spec. Top. - Accel. Beams, vol. 6, no. 2, p. 024401, Feb. 2003

Uncooled Passive THz Absorbing Pixel Designs & ROIC for Security Systems

By Alon Rotman & Roy Nicolet, Supervised by Alexander Svetlitza & Dr. Igor Brouk
TECHNION – ISRAEL INSTITUTE OF TECHNOLOGY , EE Department – VLSI Laboratory

WINNERS Kaiser EE Project Competition 2014

THz Frequency

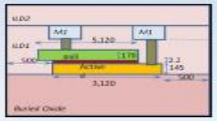


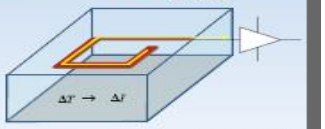
- Relatively new research area
- All objects emit THz
- High penetration: Through dust, cloth, wood, plastic and more!
- Non ionizing radiation with low intensity
- New application opportunities: Security systems, medical devices, art , car safety, etc.
- The relevant wave length is between 100-250um

TeraMOS Sensor

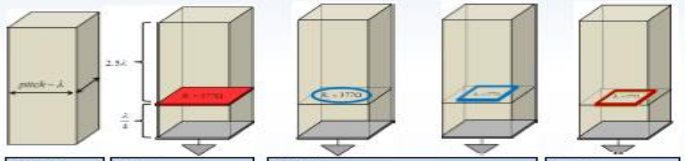
- TeraMOS is a transistor acting as thermal sensor for the THz frequency range
- The transistors main absorbing layers are made of silicon & polysilicon
- The transistor operates in the sub-threshold region
- When the radation is absorbed the transistor heats up
- The temperature change varies the current read by the readout circuit

THz radiation





Pixel Design Development



Step 1: Creating a waveguide of pitch size λ of the radiation 100-250um

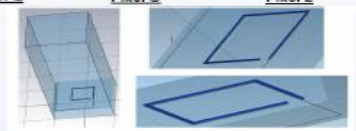
Step 2: Adding a conducting plane at the bottom and creating an absorbing layer of 377Ω at λ/4

Step 3: Creating oractical designs of a circular and square absorber to find impedance match with less thermal mass. The absorber's length is λ. Adding titanium gives the best impedance matching in ideal model

Step 4: Designing an absorber using CMOS-SOI, implementing TeraMOS sensors

Step 5: Investigating 3 designs of absorbers, Q, O & L made from silicon and polysilicon and connected to a holding arm made of silicon for conducting current from sensors. The absorption efficiency with a maximum value of 70% is not satisfactory.

Step 6: Adding a titanium coating to pixels. At a thickness of 0.2μm and 0.73 μm above silicon layers. Since titanium was proved to bring better results at the THz range as it has conductivity of 2.36×10^6 S/m. The frequency response should be simulated using an array rather than single pixel since the frequency response is different.



Current Mode ROIC with Active Feedback

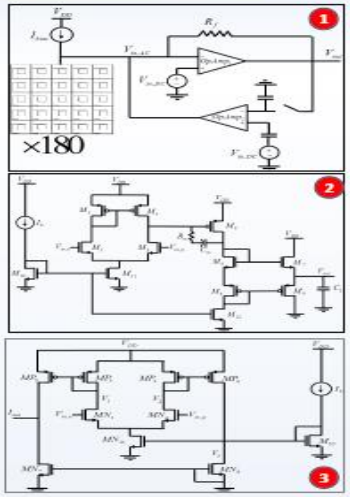
1. Current Mode ROIC
Designed for high gain, using a current source that sets the operating point for an array of pixels and a trans-impedance amplifier. There is a feedback loop with a trans-admittance amplifier for calibrating miss-match and 1/f noise and a switch +memory capacitor to prevent overriding the input signal from the pixel array.

2. Voltage-Voltage Amplifier
Implemented with a differential pair and common source stage. A buffer was added to shift the far pole beyond the omega crossover. A Miller capacitor & compensation resistor were added for stability.

3. Trans-admittance Amplifier
Differential pair and current mirror.

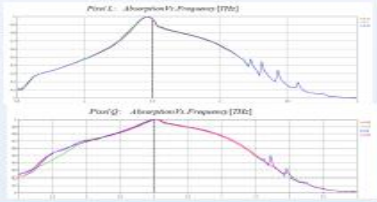
4. Switch
Transmission gate, inverter and clock pulse.

5. Pixel
One transistor, an arm as a resistor, held in sub-threshold.



Pixel Simulation Results

- **Pixel Q:** Achieves 99.99% absorption at 1THz. This device was originally designed to operate at 1.6 THz but the FSS (frequency selective surface) response moved the maxima. The outcome is still useful as 1THz is in still the required range.
- **Pixel L:** Achieves 99.99% absorption at 1.6 THz (the target frequency) with a wide bandwidth.
- **Pixel O:** Achieves 99.999% absorption at 3THz with the best band pass frequency response.

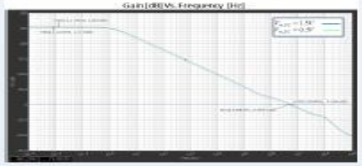


ROIC Simulation Results

ROIC to meet requirements:

- Outstanding 1/f noise suppression
- High gain and deep stability
- Low thermal noise

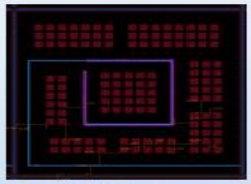
Specifications	V_{in}=0.5V	V_{in}=1.5V
Gain	128.5dB	127dB
PM	82.29°	78.8°
GBW	4.1MHz	4.5MHz
I_{DD}	0.893mA	0.954mA
V_{in} Noise	6.2n $\frac{nV}{\sqrt{Hz}}$ @ 100k	6.6n $\frac{nV}{\sqrt{Hz}}$ @ 100k
Thermal Noise	0.145 $\frac{nV}{\sqrt{Hz}}$	0.234 $\frac{nV}{\sqrt{Hz}}$



Conclusions and Achievements Pixel Design

Conclusions
- To achieve good absorption:
  - Created a waveguide with pitch of the wavelength
  - Placed absorber at λ/4 from reflecting plane
  - Use of a square shape gave the best absorbing device with FSS simulations
  - Absorbing device size should be of the same order as the wavelength of radiation
  - Titanium used to provide the best impedance match
- Extensive 3D simulation in simulator (CST) used to optimize geometric dimensions

Achievements
- The implemented absorbers are significantly superior to all previous absorbers designed in this research
- The pixels were sent for fabrication

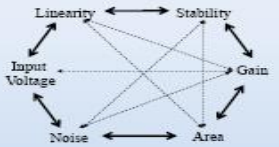


Pixel O layout

Conclusions and Achievements ROIC Design

Conclusions
- An effective ROIC was designed and implemented:
  - Current mode for higher gain
  - Momentary TA feedback for noise reduction
  - A buffer, Miller capacitor and compensation resistor added to solve stability issues.
  - Designed for low 1/f and thermal noise

Achievements
- A novel calibration technique used to overcome noise issues
- The ROIC meets and exceeds all initial requirements:
  - High gain
  - Good noise suppression
  - Deep stability



Design Tradeoffs

Hardware Scheduler for Quad Core Multiprocessor

Idan Regev and Noy Nakash supervised by Jose Yallouz
TECHNION – ISRAEL INSTITUTE OF TECHNOLOGY , EE Department – VLSI Laboratory

Multicore Processors

- The continuous drive for increased performance has led to an unsustainable increase in power density
- Multicore architectures can provide more performance without further increase in power density
- Allow increase in performance and at the same time allow a reduction in clock frequency

Multicore Challenges

- Cache Coherency

  Multiple copies of data must be kept coherent and consistent using snoopy cache protocols (MSI, MESI etc).

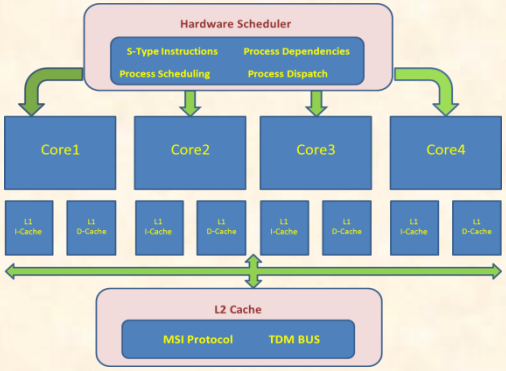
- Process Scheduling

  Efficient dispatch of processes/threads to available core according to priority, data dependencies, resources, etc.

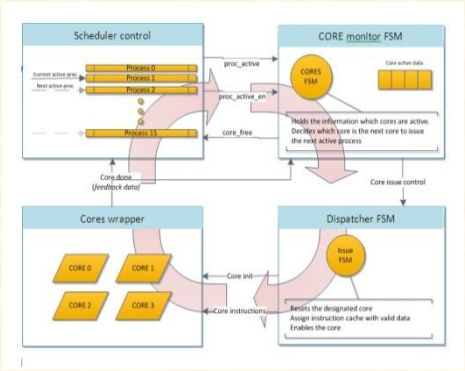
Project Goals

Replace process dependency OS software scheduling with a custom designed hardware scheduler composed of :

•Custom hardware scheduling module
•New S-Type Instructions

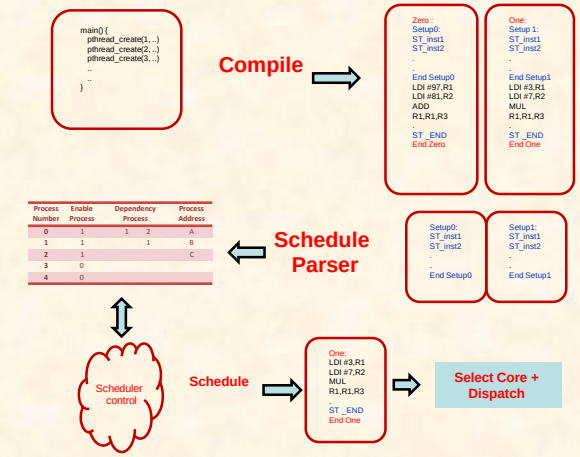


Hardware Scheduler Sub-Units



- Scheduler control FSM builds the process and processes dependencies table
  - Decides which of next process can be issued for execution
  - Can support up to 16 processes simultaneously
- Core monitor FSM maintains cores' state
- The dispatcher FSM controls the reset and initialization sequence
- The cores wrapper monitors process execution and sends process completion signal to the scheduler control FSM

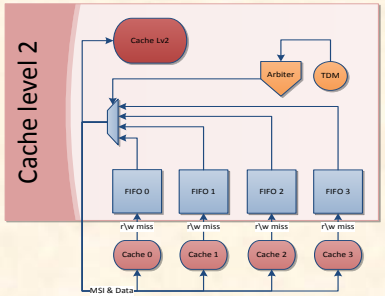
Flow Control



S-Type Instructions

#	Name	Opcode	Process id	Process rt	function
1	New process	1111	[11:8]	XXXX	0000
2	Add process dependency	1111	[11:8]	[7:4]	0001
3	Remove process dependency	1111	[11:8]	[7:4]	0010
4	Define process address	1111	Start Address		0011
5	Define process Size	1111	Number of Instructions		0100
6	End process setup	1111	[11:8]	[7:4]	1000
7	End setup	1111	[11:8]	XXXX	1111
8	Process done	1111	[11:8]	XXXX	1110

Cache Coherency

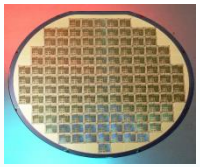


- A BUS controller with TDM arbitration synchronizes between the cores' L1 caches, L2 cache and the main memory
- The L2 cache snoops each L1 cache separately and stores all of its calls in fifos
- Inside the L2 cache an arbiter divides the time slots among the L1 caches and every clock checks the MSI and data FIFOs

Original Contributions

- The development, design and implementation of a completely new process scheduling paradigm
- Exploiting advantages of combining new S-Type instructions with a hardware scheduling module
- Defining every new S-type instruction in a way that future compilers could add them to existing code by recompilation
- Providing a process scheduler that can be expanded to perform additional tasks which are currently performed by the OS scheduler
- Design of a new architecture for the MSI cache coherency protocol for a multicore processor

TECHNION – ISRAEL INSTITUTE OF TECHNOLOGY



VLSI SYSTEMS RESEARCH CENTER

VLSI LABORATORY

DEPARTMENT OF ELECTRICAL ENGINEERING DEPARTMENT OF COMPUTER SCIENCE

Mono-Genetic Algorithm Chip for TSP

Alex Bunin and Sagi Sheer

Supervisors: Goel Samuel

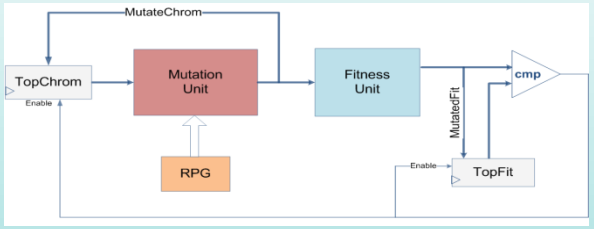
Project Goal

Design and Implement a VLSI Chip that Solves the TSP Problem using the Mono Genetic Algorithm

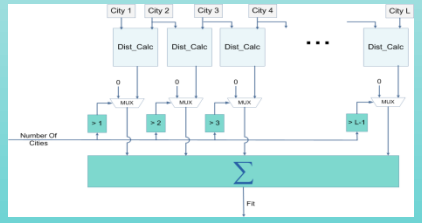
Adaptation of Genetic Algorithm to TSP

- **Define suitable representation for paths (solutions)**
- **Define mutations and how to apply them**
- **Find an efficient method of evaluating the quality of the solution**

Main Algorithm Flow

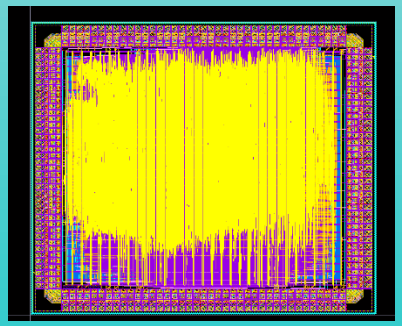


Fitness Unit



Final Layout

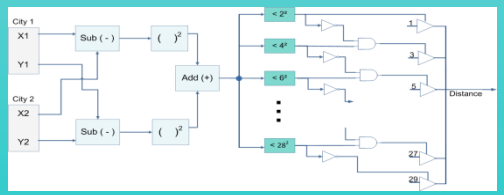
- **Technology : Tower 0.18u**
- **Frequency : 90 MHz.**
- **Area : 3 X 3 sq mm.**
- **Number of IO Pins : 160**
- **Power Consumption :300mW**



Packaged Devices

2-Cities Distance Calculation

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$



תחילת העבודה :
יש לפנות מיד למנחה על מנת לקבוע מועד לפגישה ראשונה!

אמנון סטניסלבסקי	amnon.stanislavsky@intel.com	
שרון רכניץ	sharmal@campus.technion.ac.il	5022
יניב קורמן	yanivkurman100@gmail.com	4700
שחר ג'ינו	sgino209@gmail.com	
לאוניד יביץ	leonid.yavits@nububbles.com	
עמית ברמן	bermanam@tx	
אלכס זיבאגנצב	alehan99@gmail.com	4683
ניקולס וונשטיין	nicolasw@campus.technion.ac.il	TCEפ' 408
איימן מועלם	ayman@optima-da.com	

יש לפנות מיד למנחה על מנת לקבוע מועד לפגישה ראשונה!

רון קיפר	ron@marvell.com	
רתם בן חור	rotemgby@gmail.com	
פ' אריאל אפשטיין	epsteina@ee.technion.ac.il	
עדי חנוכה	adiha@campus.technion.ac.il	
גבי לבל	gb.loebl@gmail.com	
לואי דניאל	danial.loai@gmail.com	

בהצלחה !