ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

"Σχεδιασμός και Υλοποίηση Επιταχυντή Υλικού για τον Αλγόριθμο Εκτίμησης Κίνησης Εξαντλητικής Αναζήτησης"

Θωμάς Ι. Μακρυνιώτης

Επιβλέπων: Δρ. Μηνάς Δασυγένης

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

http://arch.icte.uowm.gr

Outline

- Καθορισμός του προβλήματος
- Περιγραφή της λύσης
- Θεωρητικό υπόβαθρο
- Σχεδιασμός του συστήματος
- Λογισμικό
- Έλεγχος του συστήματος
- Αποτελέσματα - Συμπεράσματα

Καθορισμός του Προβλήματος

- Κωδικοποίηση/Συμπίεση Video
- Σύγχρονες μορφές συμπίεσης
  - Δειγματοληψία
  - Εκτίμηση Κίνησης (Motion Estimation)
  - Αντιστάθμιση Κίνησης (Motion Compensation)
- Εκτίμηση Κίνησης:
  - Η διαδικασία κατά την οποία υπολογίζεται η μεταβολή της θέσης τμημάτων ενός καρέ σε σχέση με τα επόμενα / προηγούμενα
- Εξαιρετικά Πολύπλοκη Διαδικασία!

Καθορισμός του Προβλήματος

- FSME: Full-Search Motion Estimation Algorithm (Εξαντλητική Αναζήτηση)
- Για τον υπολογισμό των διανυσμάτων κίνησης απαιτείται τεράστια υπολογιστική ισχύς → κατανάλωση ενέργειας
- Ασύμφορη η υλοποίησή του σε software για CPU γενικής χρήσης
- Συμπίεση video υψηλής ποιότητας (HD, 4K) σε πραγματικό χρόνο από συσκευές όπως smartphones, φορητές κάμερες κλπ ;

Περιγραφή της Λύσης

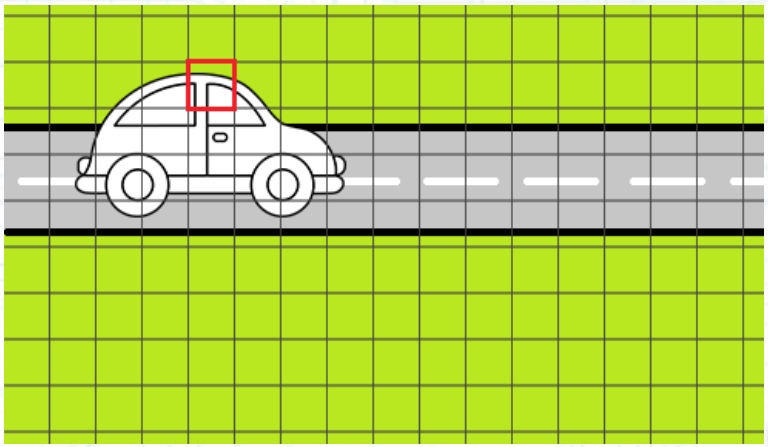
- Δημιουργία ενός κυκλώματος ειδικού σκοπού, με μοναδική λειτουργία την εκτέλεση του FSME – Χρήση VHDL
- Χρήση του κυκλώματος ως επιταχυντή / συνεπεξεργαστή σε ενσωματωμένο σύστημα
- Βελτίωση της ταχύτητας υπολογισμού των MVs σε βαθμό τέτοιο ώστε να επιτρέπεται η χρήση του για real-time κωδικοποίση

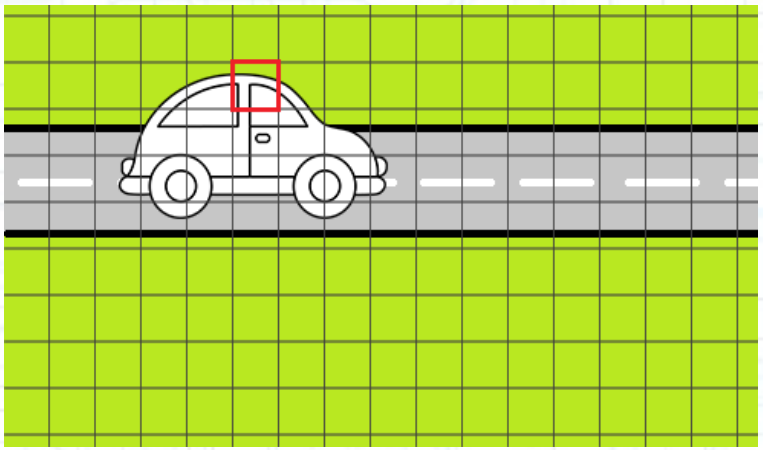
Θεωρητικό Υπόβαθρο

- Συμπίεση εικόνας
  - Συσχετισμός μεταξύ γειτονικών pixel
  - DCT (Discrete Cosine Tranform) - JPEG
  - Chrominance Downsampling
  - DPCM (Differential Pulse-Code Modulation)
  - DEFLATE – PNG
- Χωρικός Πλεονασμός (Spatial Redundancy)

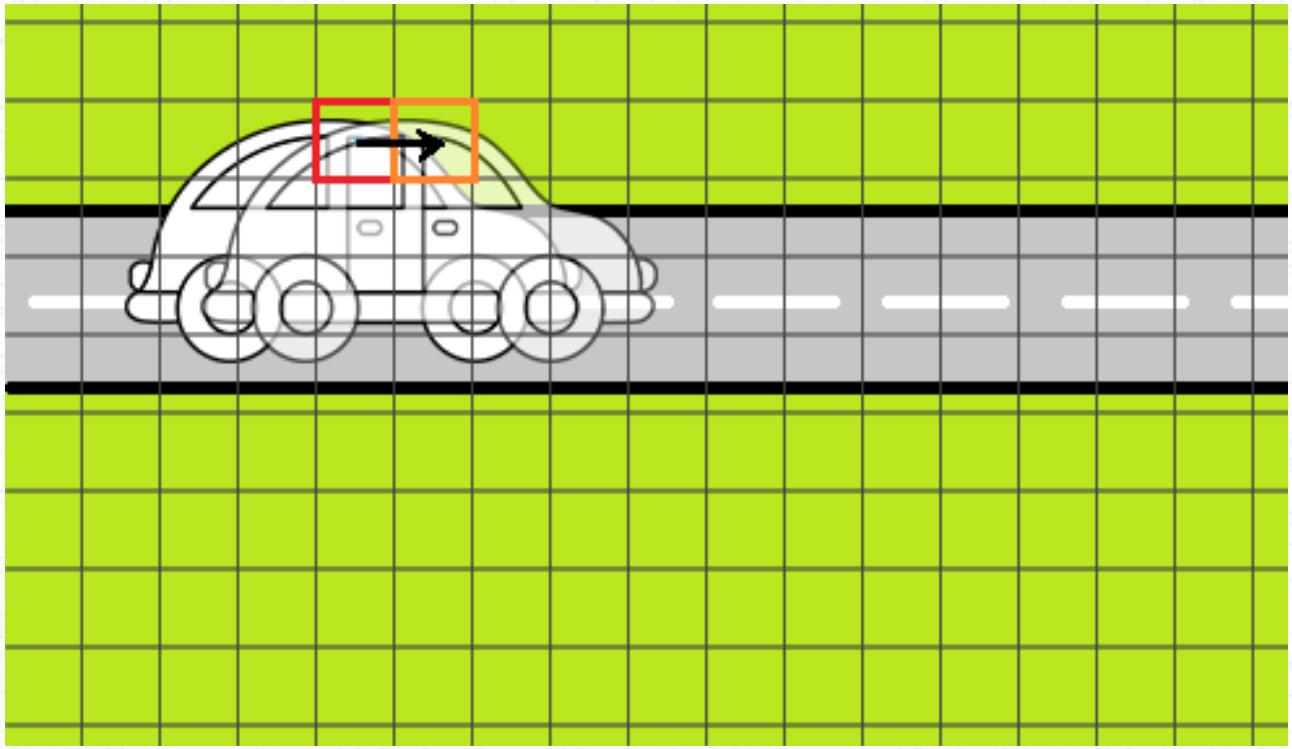
Θεωρητικό Υπόβαθρο

- Συμπίεση video
  - Κοινά χαρακτηριστικά με την εικόνα
  - Motion-JPEG
- Χρονικός πλεονασμός (Temporal Redundancy)
- Ρυθμός: Καρέ / Δευτερόλεπτο





Διάνυσμα Κίνησης



Θεωρητικό Υπόβαθρο

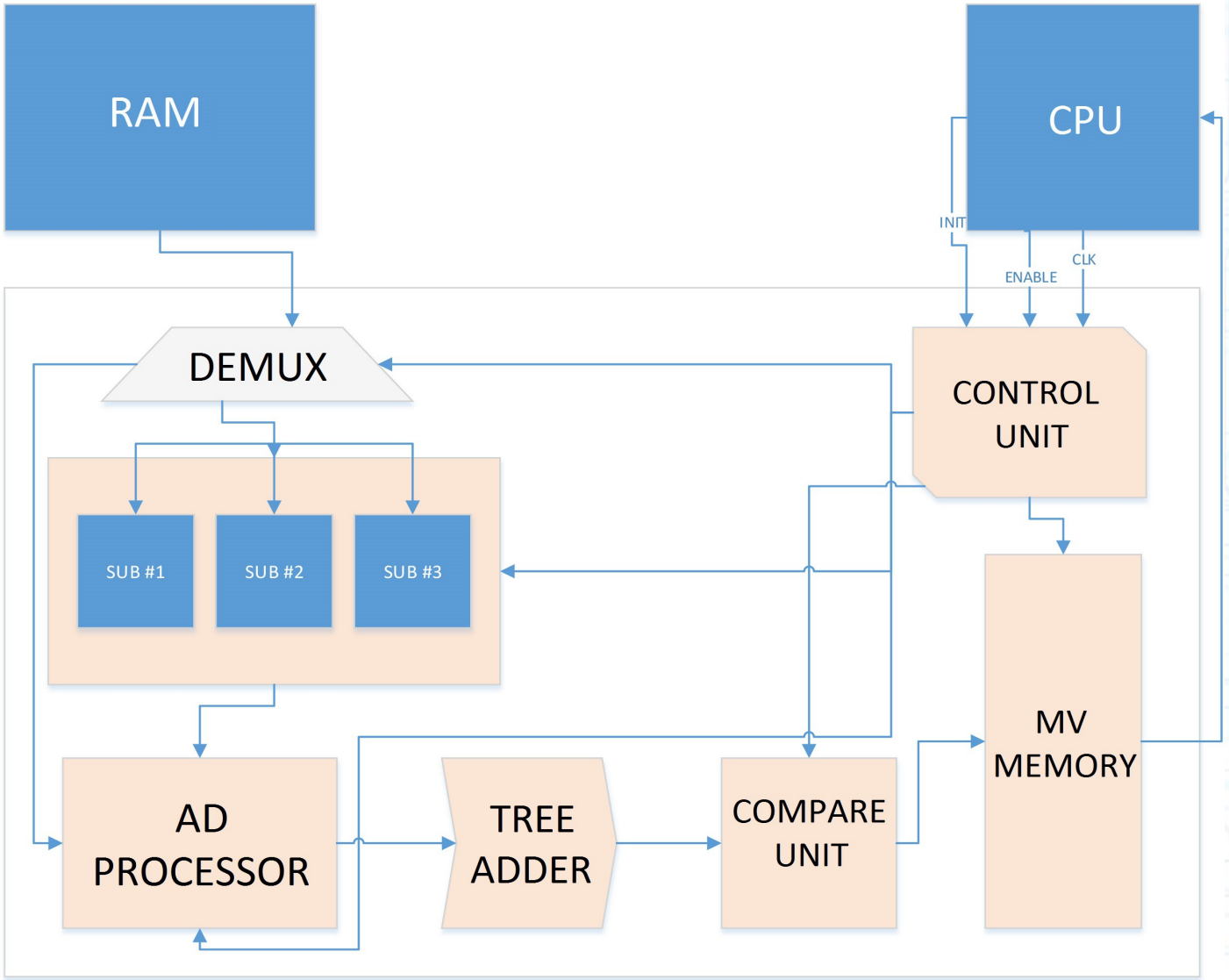
Framerate (FPS)	Αντίληψη κίνησης
10-12	Το απόλυτο ελάχιστο για την αντίληψη συνεχόμενης κίνησης. Οτιδήποτε μικρότερο, γίνεται αντιληπτό ως ξεχωριστές εικόνες.
<16	Η αλληλουχία των frames είναι ορατή. Μπορεί να ενοχλήσει πολλούς θεατές.
24	Το ελάχιστο ανεκτό για την αντίληψη μιας ομαλής κίνησης. Αποτελεί το κινηματογραφικό πρότυπο.
30	Πολύ καλύτερο σήμα σε σχέση με τα 24 FPS. Αποτελεί την καθιερωμένη πρακτική για το πρότυπο NTSC λόγω της συχνότητας του εναλλασσόμενου ρεύματος στις ΗΠΑ (60Hz), καθώς και την πλειοψηφία των ψηφιακών βίντεο.
48	Αυξημένη ποιότητα, που πλησιάζει την αντίληψη της κίνησης ως φυσική.
60	Ιδιαίτερα αυξημένη ποιότητα σε σχέση με τα προηγούμενα framerates. Οι περισσότεροι άνθρωποι αντιλαμβάνονται την κίνηση ως φυσική.

* 30 FPS → 1 Frame ανα ~33ms

Αρχιτεκτονική του Επιταχυντή

- Η βασική αρχιτεκτονική προτάθηκε στο παρελθόν από άλλους ερευνητές (Olivares et al)
- Ο επιταχυντής αποτελείται από 5 μονάδες:
  - Τοπική μονάδα μνήμης
  - Μονάδα Υπολογισμού SAD
  - Συγκριτής
  - Μνήμη Διανυσμάτων Κίνησης
  - Μονάδα Ελέγχου

Διάγραμμα της Αρχιτεκτονικής



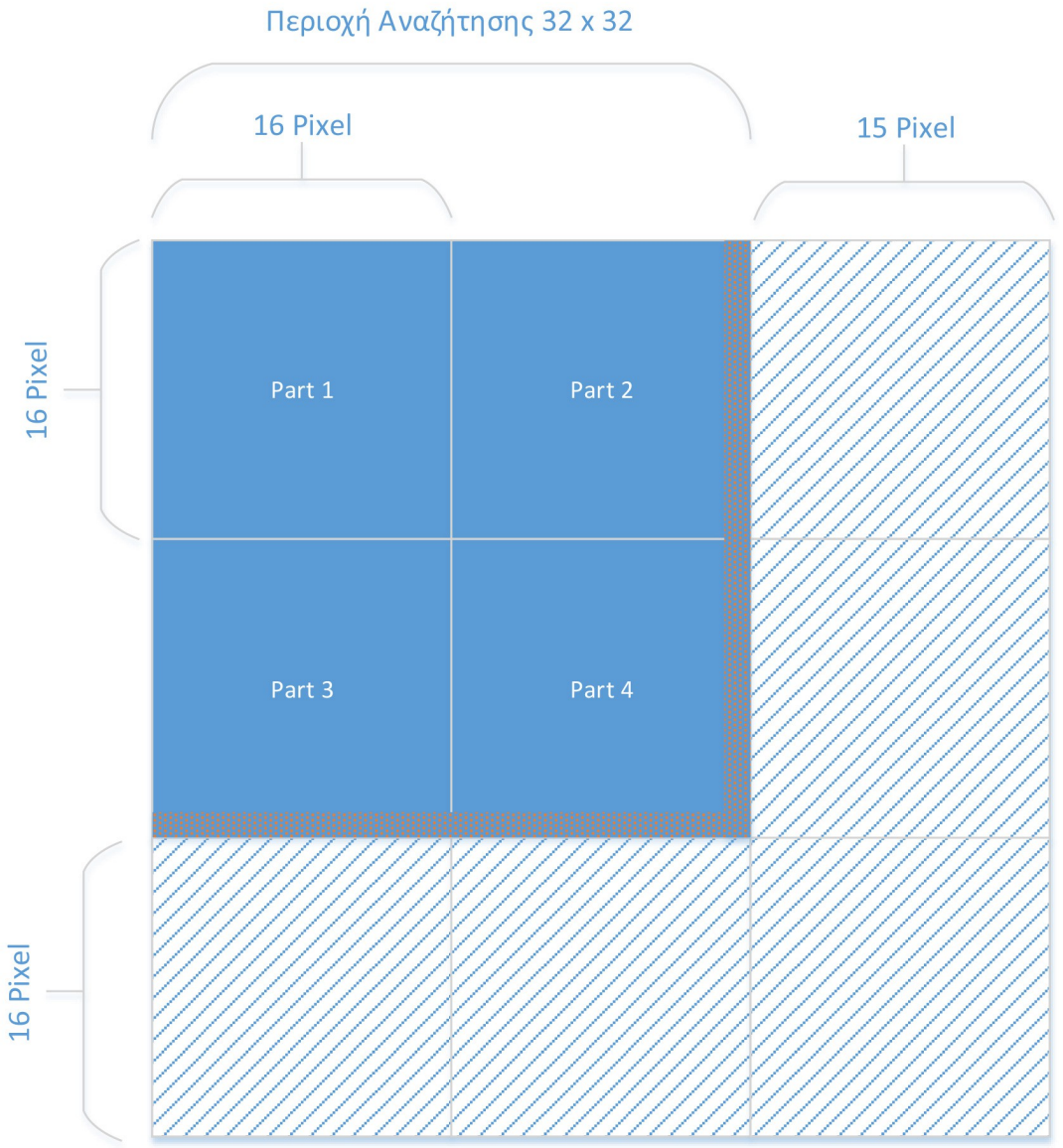
Μονάδα Τοπικής Μνήμης

- Η μονάδα τοπικής μνήμης αποτελείται από δύο υπομονάδες: τον αποπολυπλέκτη δεδομένων (data demux) και τη μονάδα μνήμης
- Demux → Δρομολογεί τα δεδομένα είτε κατευθείαν στη μονάδα SAD είτε στην τοπική μνήμη για αποθήκευση
- Μονάδα μνήμης → Αποτελείται από τρεις υπο-μνήμες και πρακτικά πρόκειται για ένα register-file

Μονάδα Τοπικής Μνήμης

- Κάθε υπομνήμη αποτελείται από ενα 16 × 16 register file
- Κάθε register έχει data width των 8-bits, οπότε, μπορεί να αποθηκεύσει την τιμή ενός pixel σε διαβαθμίσεις του γκρι (greyscale)
- Οι υπομνήμες αποθηκεύουν ένα τμήμα από την περιοχή αναζήτησης των 32 × 32 pixel
- Raster scan και σύγκριση με το current block → η υπόλοιπη περιοχή αναζήτησης φορτώνεται γραμμη-προς-γραμμη.

Περιοχή Αναζήτησης



Επαναχρησιμοποίηση Δεδομένων

- Κύριο πλεονέκτημα της αρχιτεκτονικής → **μηχανισμός επαναχρησιμοποίησης δεδομένων**
- Ο μηχανισμός προσπέλασης είναι πρακτικά ένας εσωτερικός μετρητής που επιτρέπει την προσπέλαση κάθε στήλης θέτοντας απλώς την αντίστοιχη τιμή
- Η εγγραφή και η ανάγνωσή γίνονται με έναν ιδιαίτερο τρόπο, γραμμή-προς-γραμμή και στήλη προς στήλη → Επαναχρησιμοποίηση δεδομένων χρησιμοποιώντας “έξυπνο” dataflow

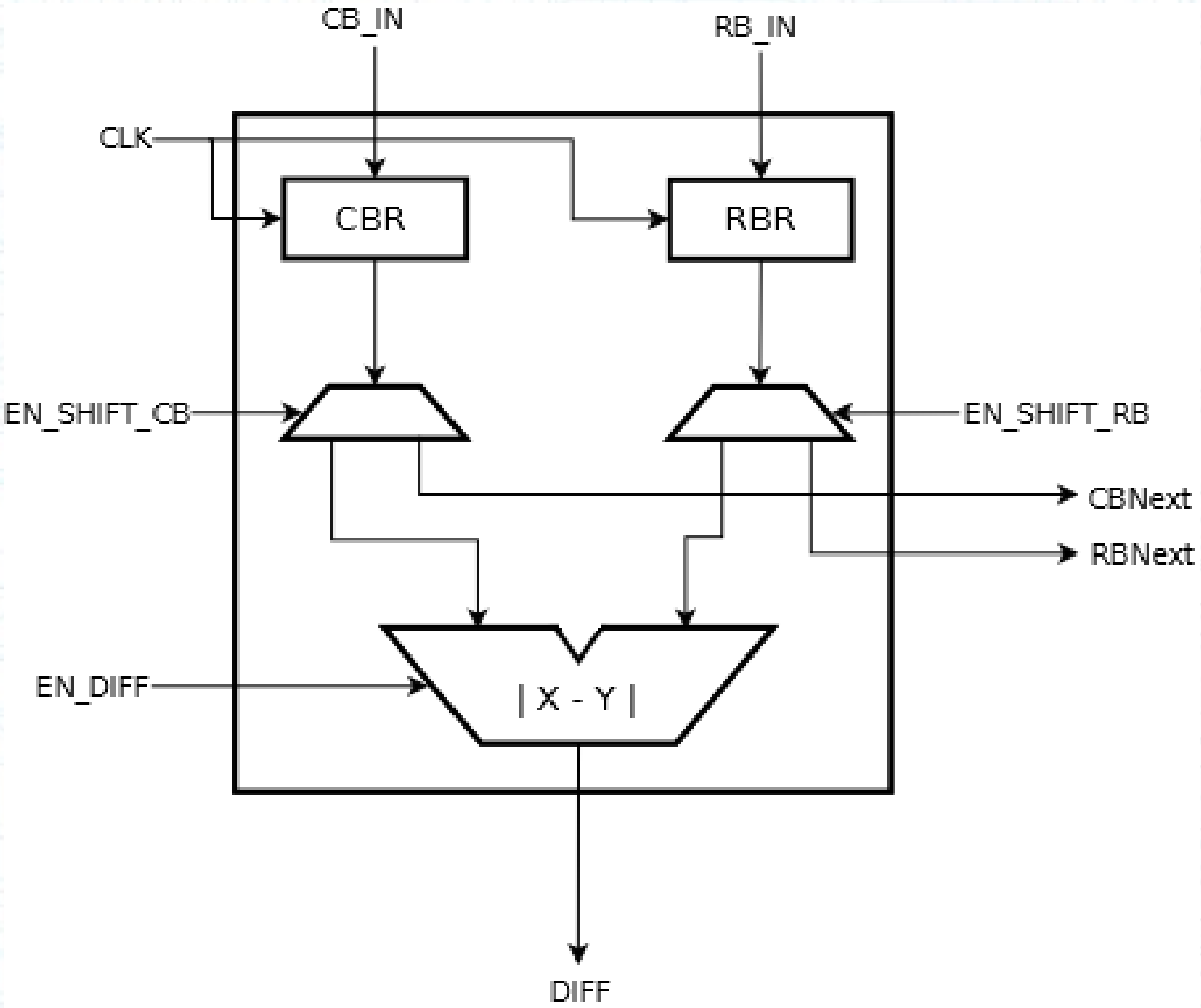
Μονάδα Υπολογισμού SAD

- Δύο διαφορετικά submodules: Absolute Difference Processor & Adder Tree
- Κριτήριο SAD:

$$SAD(i,j) = \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} | s(n_1, n_2, k) - s(n_1+i, n_2+j, k-l) |$$

- Ο AD Processor παράγει ένα concatenated διάνυσμα των 2048 bit. Οι παραχθείσες τιμές πρέπει να αθροιστούν μ'εναν πολύ γρήγορο τρόπο
- Ο Αθροιστής Δέντρου αποτελείται κυρίως από 4:2 compressors και μερικούς πλήρεις αθροιστές γενικής χρήσης

Στοιχείο Επεξεργασίας



Αθροιστής Δέντρου

- Λειτουργεί σε block των 16 x 16 (“σπάει” το διάνυσμα των 2048 bit σε ομάδες των 16 x 16 x 8-bit)
- Ο υπολογισμός κάθε block γίνεται ξεχωριστά και παράλληλα με τα υπόλοιπα
- Τελικό output: 16-bit τιμή

Μονάδα Σύγκρισης

- Η μονάδα συγκρίνει το καινούργιο SAD που παράχθηκε (και τη θέση στην οποία εντοπίστηκε) με το προηγούμενο, το οποίο βρίσκεται αποθηκευμένο σε έναν εσωτερικό καταχωρητή, μετά τη συγκριση του current block με κάποιο άλλο reference block.
- Αν η τιμή του νέου SAD < τιμή του παλιού SAD, τότε την κρατάμε, διαφορετικά απορρίπτεται.
- Η θέση του SAD υπολογίζεται από έναν εσωτερικό μετρητή που αυξάνει σε κάθε κύκλο ρολογιού.
- Output: 11-bit position signal

Μνήμη Διανυσμάτων Κίνησης

- FIFO
- 1395 11-bit registers

Μονάδα Ελέγχου

- Ιδιαίτερα περίπλοκο σύστημα, το σημαντικότερο του κυκλώματος
- Ελέγχει όλα τα σήματα που διακινούνται μέσα στον επιταχυντή
- Αποτελείται από:

  - Incremental counter

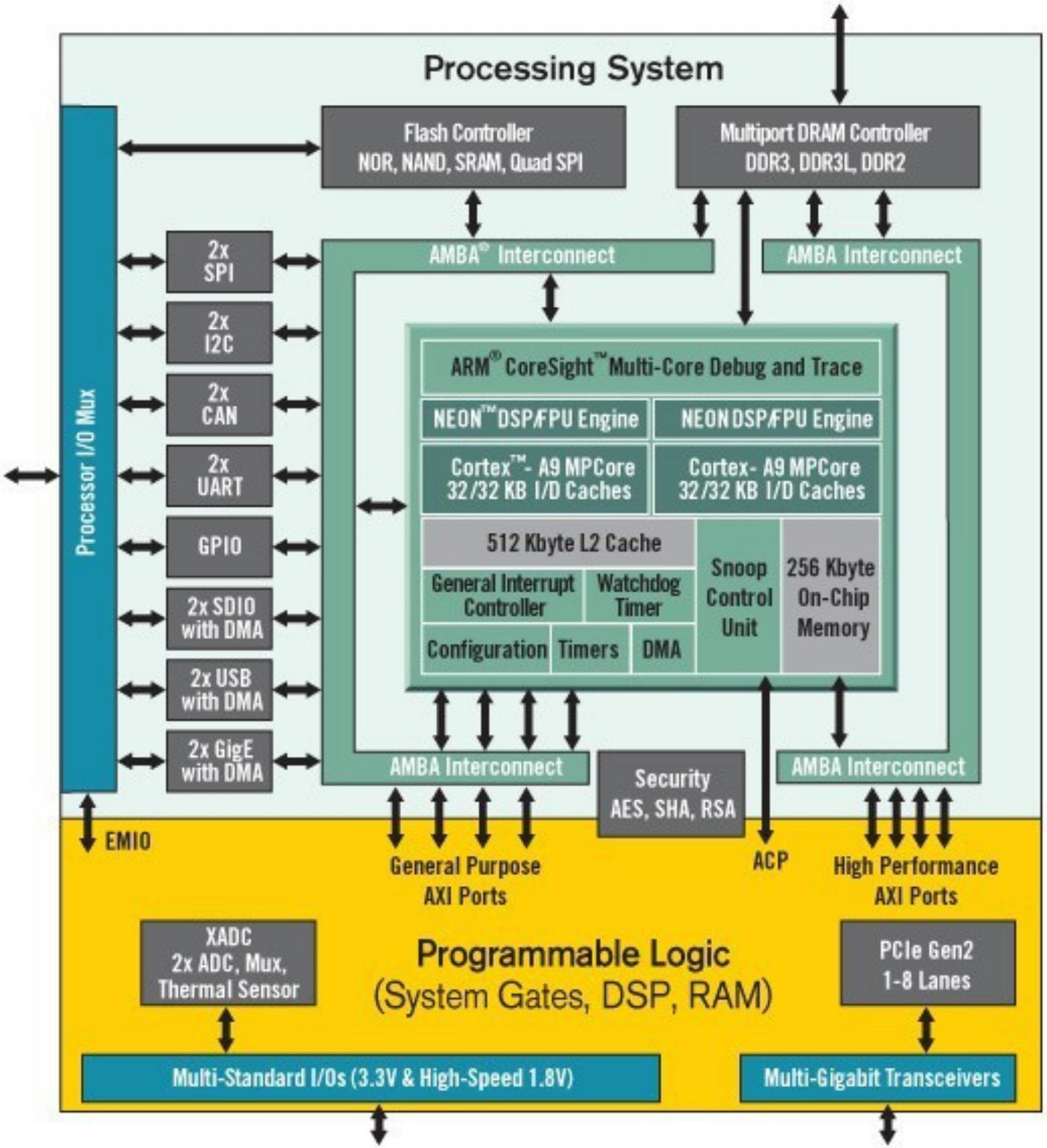
  - Signal controller
- State machine – Incremental counter αυξάνει μέχρι την τιμή 400'h πριν το reset

Αρχιτεκτονική του Ενσωματωμένου Συστήματος

- Το σύστημα που σχεδιάστηκε περιλαμβάνει τον επιταχυντή και την ARM CPU
- Περιλαμβάνει επίσης λογική που επιτρέπει την επικοινωνία μεταξύ των components και τη μεταφορά δεδομένων
- Αυτή η “glue logic” βασίζεται στο πρωτόκολλο AXI όπως όλα τα συγχρονα ARM SoC

AMBA – AXI

- Advanced Microcontroller Bus Architecture
- Το σύστημα ZYNQ χρησιμοποιεί την τρίτη γενιά του AMBA – AXI bus (AXI4)
- Στοχεύει σε high-performance, high clock frequency designs
- Κάθε περιφερειακό σχεδιασμένο για χρήση με το ZYNQ, θα πρέπει να επικοινωνεί πάνω από τον κοινό AXI bus, για βέλτιστη απόδοση


Processing System
Flash Controller
NOR, NAND, SRAM, Quad SPI
Multiport DRAM Controller
DDR3, DDR3L, DDR2
Processor I/O Mux
2x SPI
2x I2C
2x CAN
2x UART
GPIO
2x SDIO with DMA
2x USB with DMA
2x GigE with DMA
AMBA® Interconnect
AMBA Interconnect
ARM® CoreSight™ Multi-Core Debug and Trace
NEON™ DSP/FPU Engine
NEON DSP/FPU Engine
Cortex™- A9 MPCore 32/32 KB I/D Caches
Cortex- A9 MPCore 32/32 KB I/D Caches
512 Kbyte L2 Cache
General Interrupt Controller
Watchdog Timer
Configuration
Timers
DMA
Snoop Control Unit
256 Kbyte On-Chip Memory
AMBA Interconnect
Security AES, SHA, RSA
AMBA Interconnect
EMIO
General Purpose AXI Ports
ACP
High Performance AXI Ports
XADC 2x ADC, Mux, Thermal Sensor
Programmable Logic (System Gates, DSP, RAM)
PCIe Gen2 1-8 Lanes
Multi-Standard I/Os (3.3V & High-Speed 1.8V)
Multi-Gigabit Transceivers

AXI Protocols

- Υπάρχουν πολλά διαφορετικά “flavors” της διασύνδεσης AXI:

  - **AXI4**: Έκδοση για memory-mapped, high-performance IP

  - **AXI4-Lite**: Υποσύνολο του AXI4. Απλές, μονές μεταδόσεις μεταξύ memory mapped IP

  - **AXI4-Stream**: Non-memory mapped IP που απαιτούν υψηλές ταχύτητες διαμεταγωγής, και συνεχή ροή δεδομένων

AXI4 - Stream

- Ο επιταχυντής που υλοποιήσαμε είναι ακριβώς εκείνος ο τύπος IP που απαιτεί συνεχόμενη και ταχύτατη ροή δεδομένων → Το AXI4-Stream είναι μια εξαιρετική επιλογή για πρωτόκολλο διασύνδεσης.
- Χρησιμοποιεί δύο σήματα συγχρονισμού, VALID και READY
- Κύρια χρήση: DMA Controller

Direct Memory Access

- Από τη στιγμή που απαιτούμε real-time κωδικοποίηση video, η απόκριση του συστήματος πρέπει να είναι ταχύτατη.
- Για το λόγο αυτό επιλέξαμε να χρησιμοποιήσουμε μια μηχανή DMA προκειμένου να επιταχύνουμε τη διαμεταγωγή των δεδομένων
- Συγκεκριμένα χρησιμοποιήσαμε το Xilinx AXI DMA IP σε Scatter-Gather mode, προκειμένου να επιτύχουμε περαιτέρω επιτάχυνση

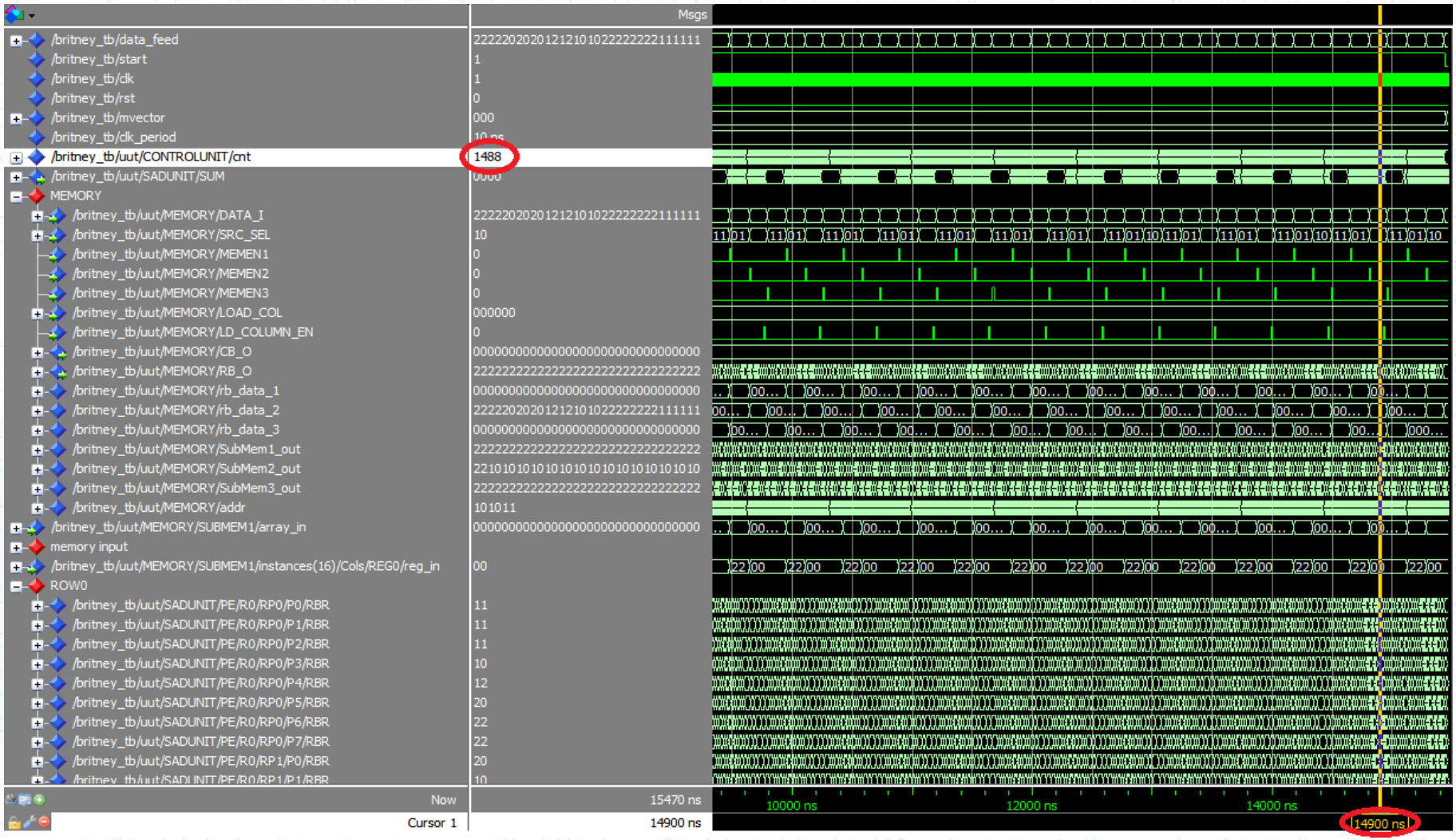
Λογισμικό

- Ελέγχος λειτουργικότητας του συστήματος → TCL Scripts & Bare metal C applications
- Παρόλα αυτα προκειμένου να δημιουργήσουμε ένα ολοκληρωμένο ενσωματωμένο σύστημα, κάναμε compile και εγκαταστήσαμε το PetaLinux μαζί με τους Xilinx DMA drivers
- User-space application → χρήση του kernel driver → επικοινωνία με IP

Έλεγχος του Συστήματος

- VHDL Testbenches
- Integrated Logic Analyzers
- TCL Scripts

ModelSim Testbenches



Αποτελέσματα

- Max op. frequency (theoretical) : 111,895 MHz

  - Απροβλημάτιστη λειτουργία στα 112 MHz
- Για μεγαλύτερα clock speeds → setup time violations
- Design area ~ 26% της συνολικής FPGA

Αποτελέσματα

XILINX VIVADO POWER REPORT

Total On-Chip Power (W)	**1,735**
Dynamic (W)	1,576
Device Static (W)	0,159
Effective TJA (C/W)	11,5
MAX Ambient (C)	65,0
Junction Temperature (C)	45,0
Thermal Margin (C)	39,7

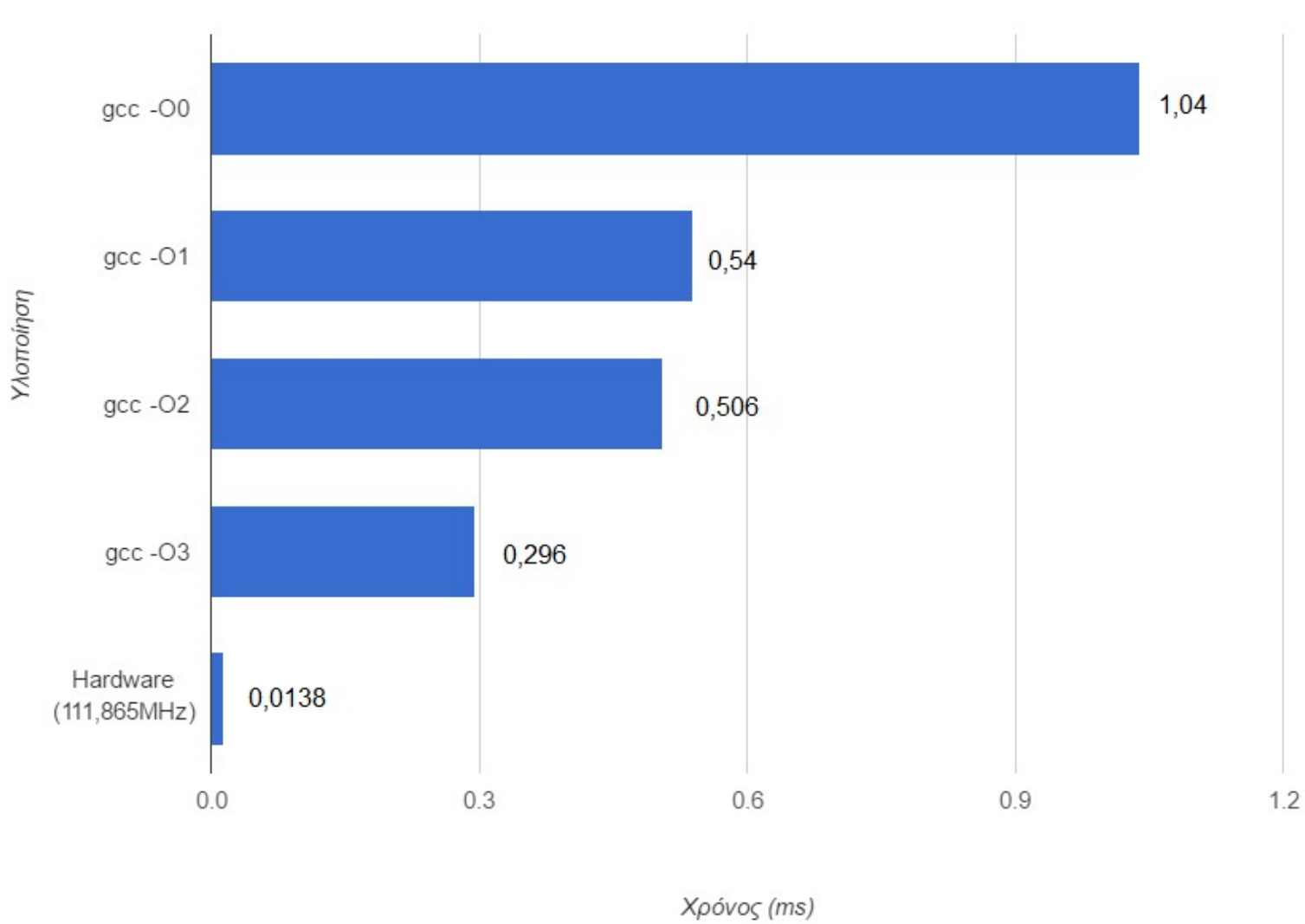
On-Chip	**Power (W)**
Clocks	0,03
Signals	0,005
Slice Logic	0,004
PS7	1,532
Static Power	0,159
Total	1,735

FPGA Area Utilization Report

Site Type	Used	Available	Utilization %
Slice LUTs	13912	53200	26,15
LUT as Logic	12041	53200	22,63
LUT as Memory	1871	17400	10,75
LUT FF Pairs	17141	53200	32,22
Slice Registers	11525	106400	10,83
Block RAM Tiles	3	140	2,14
F7 Muxes	232	26600	0,87
F8 Muxes	88	13300	0,66

- Συγκρίσιμη υλοποίηση με αυτή που προτάθηκε στο "Highspeed Motion Estimation Architecture for Real-time Video Transmission" των Goel et al
- Τα επιπλέον ~900 LUTs, οφείλονται σε άλλα συστήματα (DMA, AXI controllers, etc)

Επιτάχυνση



Η υλοποίηση στο hardware είναι πάνω από 21 φορές γρηγορότερη σε σχέση με τη βέλτιστη υλοποίηση σε software

Συμπεράσματα

- Ο σχεδιασμός και η υλοποίηση αυτού του ενσωματωμένου συστήματος απέδειξε τη δυνατότητα σχεδιασμού **ετερογενών συστημάτων υψηλών επιδόσεων**.
- Χρήση της ισχύος των νέων SoCs για την κατασκευή ακόμα καλύτερων ψηφιακών συστημάτων με πολύ μεγαλύτερες δυνατότητες.
- Επιταχυντές → Πολύ πιθανό ενδεχόμενο για το μέλλον του everyday computing, ειδικά σε processing-intensive applications

Μελλοντικές Επεκτάσεις

- Σχεδιασμός και υλοποίηση πολυπλοκότερων, εξειδικευμένων ενσωματωμένων συστημάτων, με έμφαση στη χαμηλή κατανάλωση ισχύος
- Implement greater level of data reuse
- Σχεδιασμός ενός upscaled version, με στόχο τα Ultra-High Definition (HEVC) encoding standards.

That's all Folks!

- Ο κώδικας και το documentation του project βρίσκονται στο:

  https://github.com/tommakrin/Brenda

Ευχαριστώ για την προσοχή σας!