

Universitatea Tehnică din Cluj-Napoca

Structura sistemelor de calcul

Redimensionare de Imagini

Bota Alex Ionuț

Grupa 30237

Îndrumător: Mircea Paul Mureșan

Data: 8.1.2024

1. Rezumat	3
2. Introducere.....	4
3. Fundamentare teoretică	5
4. Proiectare și Implementare.....	7
4.1. Metoda Experimentală	7
4.2. Soluția Aleasă.....	7
4.3 Arhitectura sistemului	8
4.4. Algoritmi Implementați	8
4.5 Manual de utilizare.....	9
5. Rezultate Experimentale	10
6. Concluzii	12
7. Bibliografie.....	13

1.Rezumat

Tehnologia FPGA reprezintă o soluție modernă pentru accelerarea operațiunilor de procesare intensivă, fiind utilizată într-o gamă variată de aplicații precum inteligența artificială, procesarea imaginilor și transmisii video. Cu tendințele actuale de creștere a volumului de date și a rezoluțiilor imaginilor, nevoia pentru soluții eficiente de prelucrare a datelor este mai mare ca oricând.

Proiectul de redimensionare a imaginilor utilizând platforma PYNQ-Z1 a demonstrat aplicabilitatea accelerării hardware pentru procesarea imaginilor grayscale. Tema proiectului a fost axată pe reducerea dimensiunilor imaginilor 4K la o rezoluție Full HD (1920x1080) utilizând atât resursele hardware ale logicii programabile (PL), cât și procesarea software (PS). Proiectul a integrat utilizarea bibliotecilor Python pentru manipularea imaginilor, împreună cu o interfață pentru transfer eficient de date între PS și PL. Testele efectuate au evidențiat avantajele accelerării hardware, cu timpi de execuție semnificativ reduși comparativ cu procesarea software. Concluziile subliniază eficiența utilizării FPGA-urilor pentru operațiuni de procesare intensivă, oferind oportunități pentru aplicații avansate în prelucrarea imaginilor.

2.Introducere

Proiectul prezentat se concentrează pe utilizarea platformei PYNQ-Z1 pentru redimensionarea imaginilor grayscale, evidențiind beneficiile combinării resurselor software (PS) și hardware (PL).

System on Chip (SoC) reprezintă o arhitectură care integrează multiple componente hardware pe un singur cip, cum ar fi procesorul, memoria, perifericele și FPGA. Acest design permite realizarea de sisteme complexe și eficiente energetic, fiind utilizat pe scară largă în aplicații embedded, robotică, telecomunicații și procesarea semnalelor.

O **image digitală** este o reprezentare a unei imagini reale bidimensionale (image în "2D" sau două dimensiuni), ca o mulțime finită de valori digitale (numerice).

Un **FPGA (Field Programmable Gate Array)** este un circuit integrat digital configurabil, de către utilizator, după ce a fost fabricat. FPGA este un tip de circuit logic programabil. Configurarea FPGA se face, în general, cu ajutorul unui limbaj de descriere hardware, de exemplu: HDL, VHDL.

Domeniul de studiu al proiectului este procesarea imaginilor, un subdomeniu esențial al ingineriei și științei calculatoarelor. Prelucrarea imaginilor implică transformarea datelor din imagini în forme utilizabile sau mai ușor de interpretat. Redimensionarea imaginilor, un aspect fundamental al acestui domeniu, are aplicații largi în domenii precum streaming video, vizualizare medicală și robotică. Utilizarea FPGA-urilor permite implementarea unor algoritmi de procesare rapidă, reducând latențele și consumul de energie comparativ cu soluțiile bazate exclusiv pe software.

Problema abordată în acest proiect este redimensionarea imaginilor 4K(3840x2160) la rezoluții Full HD (1920x1080) utilizând platforma PYNQ-Z1. Obiectivul principal a fost de a demonstra eficiența utilizării hardware-ului configurabil (PL) pentru această sarcină și de a compara performanțele cu procesarea software (PS). Proiectul a implicat transferul datelor printr-o interfață DMA, prelucrarea lor în logica programabilă și afișarea rezultatelor pe un ecran HDMI. S-a acordat o atenție deosebită analizei timpilor de procesare și gestionării eficiente a datelor între PS și PL.

Soluția propusă constă în utilizarea unui overlay hardware personalizat pentru PYNQ-Z1, care integrează un modul DMA și logică de redimensionare în PL. Datele

sunt transferate eficient între PS și PL prin intermediul DMA, iar imaginea rezultată este afișată atât pe HDMI, cât și salvată local. Această soluție este superioară implementărilor software prin timpi reduși de execuție și eficiență energetică.

Structura raportului este următoarea: Secțiunea "Fundamentare Teoretică" detaliază conceptele teoretice și tehnologiile utilizate. Secțiunea "Implementare" descrie pașii pentru configurarea platformei, implementarea codului și integrarea componentelor hardware și software. Secțiunea "Rezultate" prezintă analiza performanței și comparațiile dintre soluțiile hardware și software. În final, "Concluziile" sintetizează rezultatele și oferă perspective pentru extinderea proiectului.

3. Fundamentare teoretică

PYNQ-Z1 este o platformă de dezvoltare bazată pe SoC-ul **Xilinx Zynq-7000**. Acest SoC combină:

- **Un procesor ARM Cortex-A9 dual-core:** pentru sarcini generale de procesare (PS - Processing System).
- **Logica programabilă FPGA (PL - Programmable Logic):** care permite accelerarea hardware a aplicațiilor.

Placa PYNQ-Z1 este proiectată pentru a facilita dezvoltarea aplicațiilor embedded folosind limbajul **Python**. Printre caracteristicile sale cheie se numără:

- **Memorie DDR3 de 512 MB:** partajată între PS și PL.
- **Interfețe I/O:** GPIO, UART, I2C, SPI, HDMI.
- **Suport pentru extensii:** conectori Pmod și Arduino pentru adăugarea de periferice externe.

Această platformă este excelentă pentru aplicații precum prelucrarea imaginilor, învățarea automată și dezvoltarea sistemelor embedded, datorită capacității sale de a combina ușurința programării software cu puterea accelerării hardware.

Există desigur o infinitate de forme și mărimi de imagini, aceasta fiind valabil atât pentru imaginile tradiționale, analogice, cât și pentru cele digitale. Cel mai des folosite sunt însă formele dreptunghiulare, care se reflectă și la formele ecranelor de cinema, TV, calculatoare și altele. Forma imaginilor digitale dreptunghiulare se exprimă în pixeli, și anume prin numărul de pixeli pe orizontală și verticală, sau pe lățime și lungime (înălțime). Raportul dintre lățime și lungime (înălțime) se numește

de obicei „formatul” imaginii digitale. Unele formate des întâlnite sunt reprezentate în imaginea de mai jos.



Există numeroase programe care pot face ca o imagine digitală din computerul unde a fost ea stocată să devină vizibilă și pentru om. Astfel, imaginile de tipul GIF, JPEG și PNG pot fi prezentate pe un display în mod foarte simplu.

PYNQ oferă o interfață Python pentru a permite overlay-urilor din PL să fie controlate din limbajul de programare Python care rulează în PS.

Un FPGA (Field Programmable Gate Array) permite implementarea hardware a algoritmilor, oferind paralelism și latență redusă comparativ cu soluțiile software. În cazul de față, redimensionarea imaginii s-a realizat prin metoda **average pooling**, care presupune gruparea valorilor pixelilor învecinați din matricea de intrare și calcularea unei valori medii pentru generarea imaginii de ieșire. Aceasta este o metodă simplă, dar eficientă pentru reducerea dimensiunii imaginilor.[1]

Pentru transferul datelor între sistemul de procesare (PS) și logica programabilă (PL), s-a utilizat interfața **DMA (Direct Memory Access)**. DMA permite transferuri de date rapide și eficiente.[2]

Spre deosebire de soluțiile software, implementarea hardware asigură performanțe superioare, fiind potrivită pentru aplicații în timp real.

4. Proiectare și Implementare

Implementarea proiectului a fost realizată în mai multe etape, acoperind atât partea hardware, cât și cea software. În continuare, sunt detaliate aceste etape:

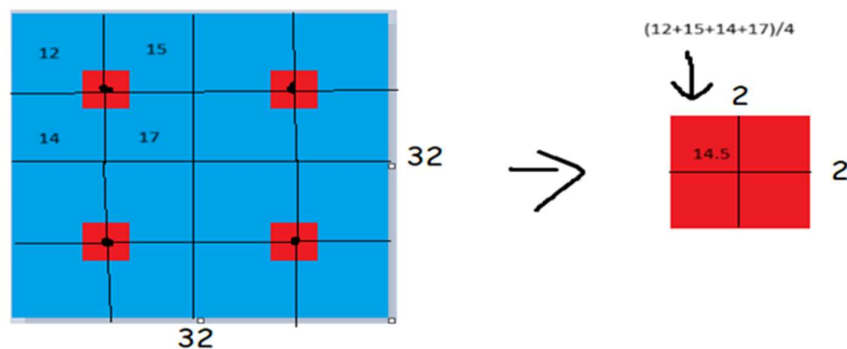
4.1. Metoda Experimentală

Metoda utilizată a implicat o combinație de implementare hardware și software:

- **Hardware:** Modul de redimensionare a fost implementat în logica programabilă (PL) utilizând Vivado și limbajul VHDL pentru a genera overlay-ul FPGA.
- **Software:** Python scris și executat în Jupyter Notebook a fost utilizat pentru configurarea platformei, transferul datelor și gestionarea afișării rezultatelor.

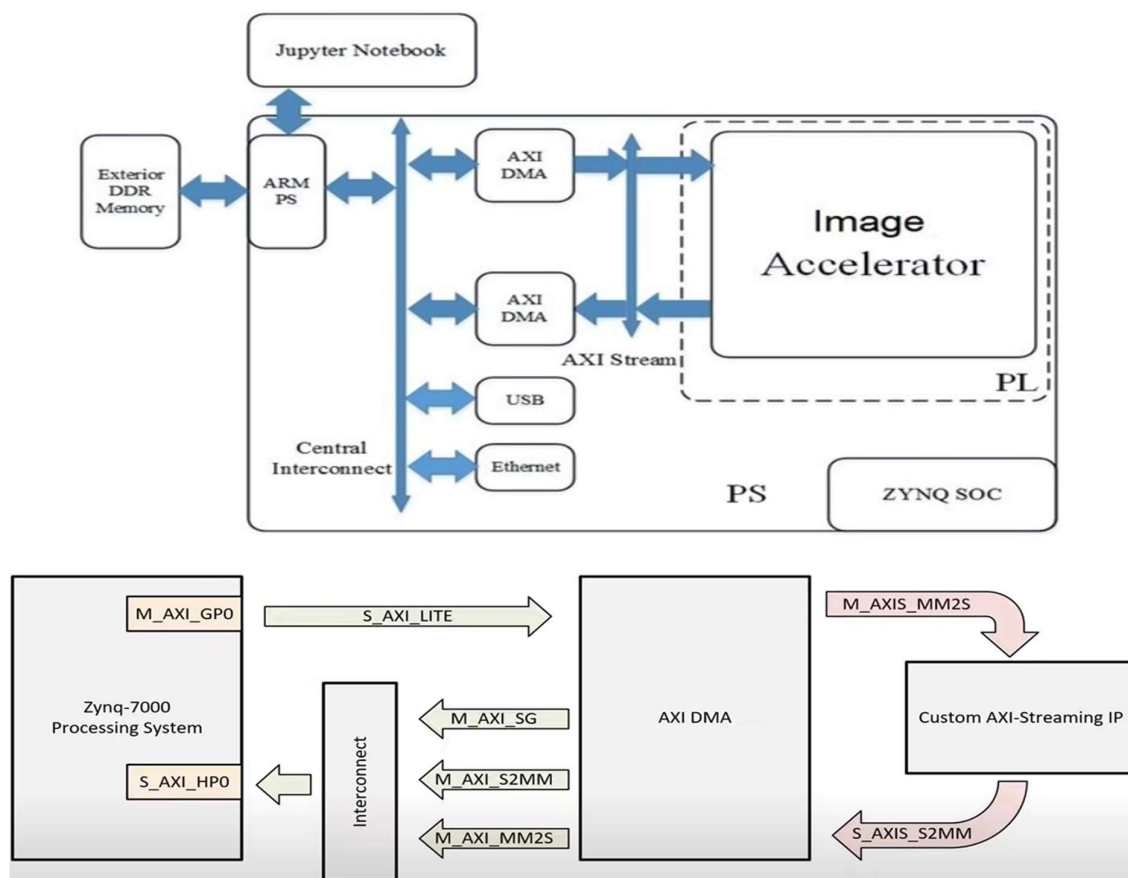
4.2. Soluția Aleasă

- Din mai multe metode de redimensionare, s-a ales algoritmul average pooling datorită simplității sale și ușurinței de implementare în hardware. Metodele mai complexe precum bilinear interpolation nu au fost alese din cauza complexității implementării hardware.



- Pentru transferul datelor de la PS la PL și invers s-a folosit DRAM-ul plăci de dezvoltare în detrimentul celulelor BRAM, factorul principal ce a influențat aceasta decizie fiind dimensiune mare a datelor ce trebuie transmise. Acest lucru permite ca datele să fie citite din DRAM și trimise la un flux AXI sau primite dintr-un flux și scrise în DRAM.

4.3 Arhitectura sistemului



Această diagramă ilustrează arhitectura de comunicare între Sistemul de Procesare Zynq-7000 (PS - Processing System) și Logica Programabilă (PL - Programmable Logic). Componentele principale ale sistemului sunt:

- Sistemul de Procesare Zynq-7000 reprezintă partea PS a plăcii, care conține procesorul ARM. Acesta comunică cu restul sistemului prin intermediul mai multor interfețe AXI:
 - M_AXI_GP0: o interfață de uz general pentru control
 - S_AXI_HP0: o interfață de înaltă performanță pentru transfer de date
- Blocul de Interconectare servește ca punct central de rutare pentru toate semnalele AXI între PS și PL.
- AXI DMA (Direct Memory Access) este un controlor specializat care gestionează transferurile eficiente de date între memorie și componentele de streaming. Acesta utilizează:
 - S_AXI_LITE pentru configurare și control
 - M_AXIS_MM2S pentru citirea datelor
 - S_AXIS_S2MM pentru scrierea datelor procesate

Pentru programarea procesorului plăcii în Python s-au utilizat:

- Biblioteca Pillow (PIL): Utilizată pentru manipularea imaginilor (citire, conversie, afișare, salvare).
- Biblioteca NumPy: Folosită pentru operațiuni matriceale asupra datelor imaginii.
- Clasa Overlay: este utilizată pentru a încărca overlay-urile PYNQ în PL și pentru a gestiona și controla overlay-urile existente. Clasa este instanțiată cu fișierul .bit pentru un overlay. În mod implicit, fișierul Tcl va fi analizat, iar fluxul de biți va fi descărcat în PL. Aceasta înseamnă că pentru a utiliza clasa de suprapunere, trebuie furnizate un .bit și .tcl și un .hwh pentru un overlay.

4.5 Manual de utilizare

1. Pregătirea cardului microSD: Începeți prin descărcarea imaginii sistemului de operare PYNQ de pe site-ul oficial (pynq.io). După descărcare, folosiți un program precum Etcher sau Win32 Disk Imager pentru a scrie imaginea pe un card microSD cu capacitate minimă de 16GB. Introduceți cardul microSD în slotul dedicat al plăcii PYNQ-Z1.
2. Conectarea hardware: Conectați placa PYNQ-Z1 la calculator folosind un cablu micro-USB pentru alimentare și programare. Asigurați-vă că jumperii de pe placă sunt configurați corect pentru boot-are de pe cardul microSD. De asemenea, conectați un cablu Ethernet între placa PYNQ și rețeaua locală pentru acces la interfața Jupyter.
3. Pornirea și accesarea sistemului: După alimentarea plăcii, așteptați aproximativ un minut pentru inițializarea sistemului. LED-ul de alimentare ar trebui să fie aprins constant, iar LED-ul "DONE" să se aprindă după finalizarea boot-ării.
4. Accesarea Jupyter Notebook: Deschideți browserul web și navigați la adresa <http://pynq:9090> sau găsiți adresa IP a plăcii în rețeaua locală și accesați-o pe portul 9090. La prima conectare, veți fi întâmpinat de interfața Jupyter Notebook, unde vi se va solicita parola (implicit: xilinx).
5. Verificarea funcționalității: După autentificare, veți avea acces la notebooks-urile demonstrative incluse în imagine. Puteți testa funcționalitatea sistemului rulând notebook-ul "Getting_Started" din directorul principal.

Pentru asistență suplimentară, puteți consulta documentația oficială PYNQ:
<https://pynq.readthedocs.io>

5. Rezultate Experimentale

Implementarea proiectului s-a realizat utilizând următoarele instrumente de proiectare:

- Limbaj de programare: Python, VHDL.
- Mediu software: Biblioteca PYNQ Python API, Pillow (PIL) pentru manipularea imaginilor, NumPy pentru operațiuni matriceale.
- Vivado Design Suite 2020.1, utilizat pentru proiectarea FPGA.
- Platformă hardware: PYNQ-Z1, bazat pe Xilinx Zynq-7000 SoC, cu procesor dual-core ARM Cortex-A9 și FPGA.
- Sistem de operare: PYNQ Linux.

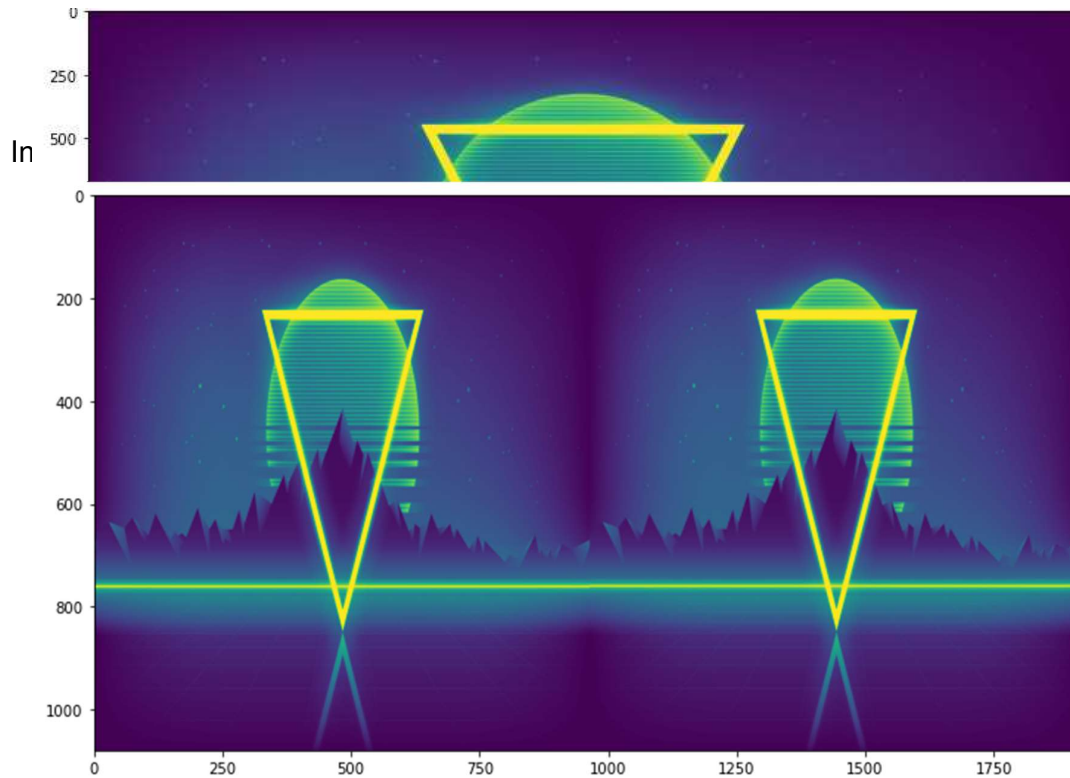
Testarea sistemului s-a realizat utilizând următoarele etape:

1. Citirea imaginilor: Imaginile de intrare au fost citite de pe cardul SD în PS și convertite la format grayscale.
2. Transferul datelor: Datele imaginii de intrare au fost trimise către PL prin DMA pentru redimensionare.
3. Verificarea datelor procesate: Imaginea redimensionată a fost returnată către PS și afișată prin HDMI pe un display extern.
4. Măsurători: Timpii de execuție pentru redimensionarea hardware (PL) și software (PS) au fost comparați.

Pentru redimensionarea unei imagini 4K (3840x2160) la Full HD (1920x1080), s-au obținut următoarele rezultate:

<i>Metodă</i>	<i>Timp de Procesare</i>	<i>Observații</i>
Hardware (PL)	0.16 secunde	Performanță ridicată
Software (PS)	1.14 secunde	Consum mare de resurse CPU

Imagine înainte:



Comparații:

- Eficiență: Redimensionarea hardware a fost de aproximativ 10 ori mai rapidă decât procesarea software.
- Resurse hardware: Implementarea în PL a consumat o cantitate mică de resurse FPGA.

Dificultăți Întâlnite:

- Sincronizarea transmiteri datelor între PS și PL.
- Calibrarea parametrilor HDMI: A fost necesară ajustarea rezoluției la (1280, 720) HD pentru afișarea corectă pe display.

6. Concluzii

Proiectul a rezolvat problema redimensionării rapide a imaginilor mari utilizând platforma PYNQ-Z1.

Avantaje:

- Performanță ridicată datorită accelerării hardware.

- Consum redus de resurse hardware, lăsând loc pentru alte funcționalități.

Dezavantaje:

- Limitarea la imagini grayscale.

Proiectul poate fi utilizat în aplicații precum:

- Prelucrare video în timp real.
- Sisteme embedded pentru robotică și vehicule autonome.
- Optimizarea imaginilor pentru diverse website-uri.

Propuneri pentru Dezvoltări Viitoare

- Extinderea suportului pentru imagini color (RGB).
- Implementarea altor metode de redimensionare, precum bilinear interpolation.

7. Bibliografie

- [1] Chaubey, Aashish. "Downsampling and Upsampling of Images — Demystifying the Theory." *Medium*, 14 January 2020, <https://medium.com/analytics-vidhya/downsampling-and-upsampling-of-images-demystifying-the-theory-4ca7e21db24a>. Accessed 6 January 2025.
- [2] Pynq. https://pynq.readthedocs.io/en/v2.4/overlay_design_methodology/pspl_interface.html.
- dimiterk. "Image processing on ZC702 using PYNQ." <https://community.element14.com/technologies/fpga-group/b/blog/posts/image-processing-on-zc702-using-pynq>.
- Fpgadeveloper. "creating-a-custom-axi-streaming-ip-in-vivado." <https://www.fpgadeveloper.com/2017/11/creating-a-custom-axi-streaming-ip-in-vivado.html/>.
- Jain, Sandeep. "How to Convert images to NumPy array?" *GeeksforGeeks*, 30 September 2022, <https://www.geeksforgeeks.org/how-to-convert-images-to-numpy-array/>. Accessed 6 January 2025.

- Jain, Sandeep. "How to Convert images to NumPy array?" *GeeksforGeeks*, 30 September 2022, <https://www.geeksforgeeks.org/how-to-convert-images-to-numpy-array/>. Accessed 6 January 2025.
- "Python Lists VS Numpy Arrays." *GeeksforGeeks*, 25 August 2023, <https://www.geeksforgeeks.org/python-lists-vs-numpy-arrays/>. Accessed 6 January 2025.