



# Projet de bachelor

Bruteforce Password Attack on FPGAs

# Table des matières :



- Introduction
- Bcrypt
- Réalisations
- Résultats
- Conclusion

---

# Introduction

# Introduction - Elca Security



# Introduction - Durée du travail



10.2023



03.2024

## Projet de semestre :

- En parallèle des cours
- 8 h par semaine



05.2024

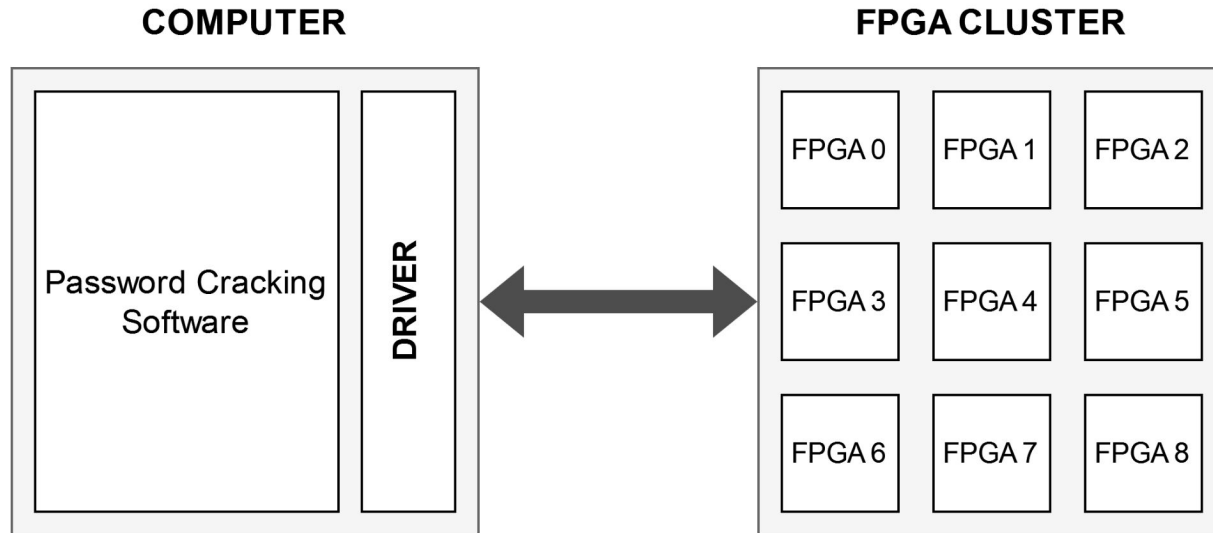


08.2024

## Projet de Bachelor

- Temps plein
- 450 h de travail

# Introduction - Schéma



# Introduction - FPGA vs CPU vs GPU



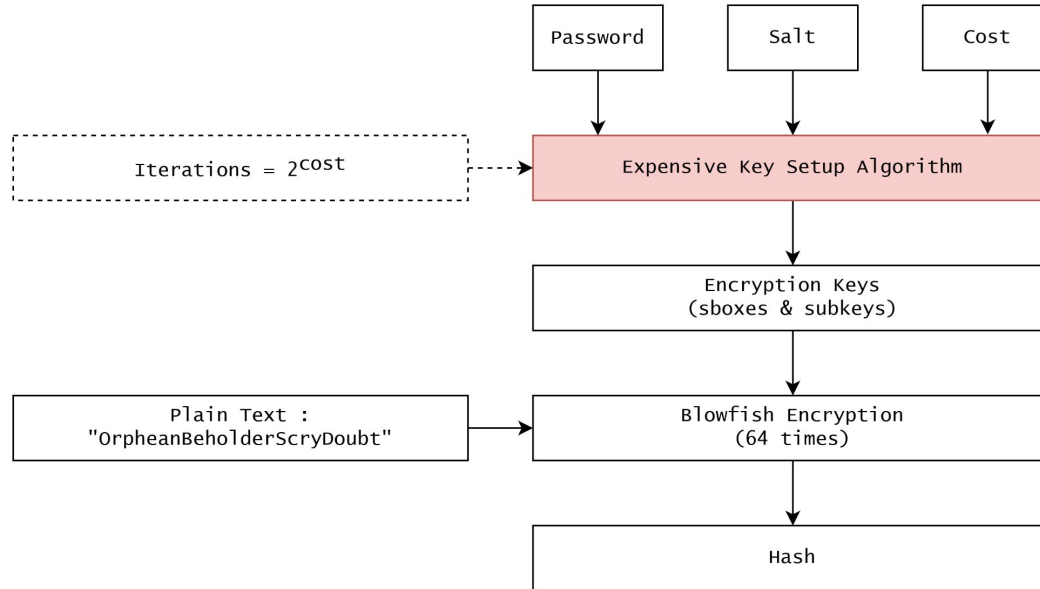
- Consommation
- Hashrate
- Coût

---

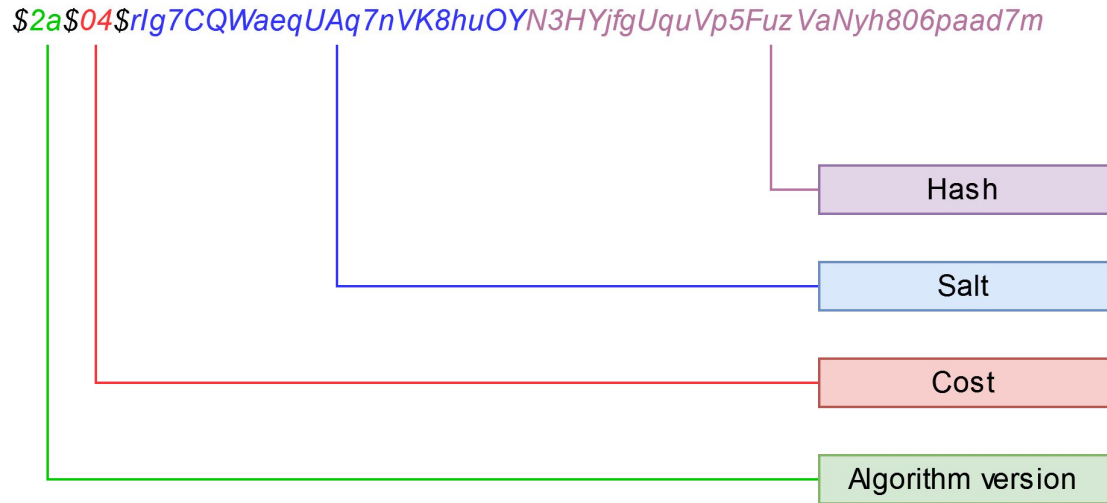
# Bcrypt - Algorithme de hash



# Bcrypt



# Bcrypt - Format du hash



---

# Réalisations - Travail de semestre

# Implémentation existante

## rub-hgi/high-speed\_bcrypt



VHDL implementation and LaTeX source of "High-Speed Implementation of bcrypt Password Search using Special-Purpose Hardware", published at ReConFig'14



1

Contributor



0

Issues



1

Star



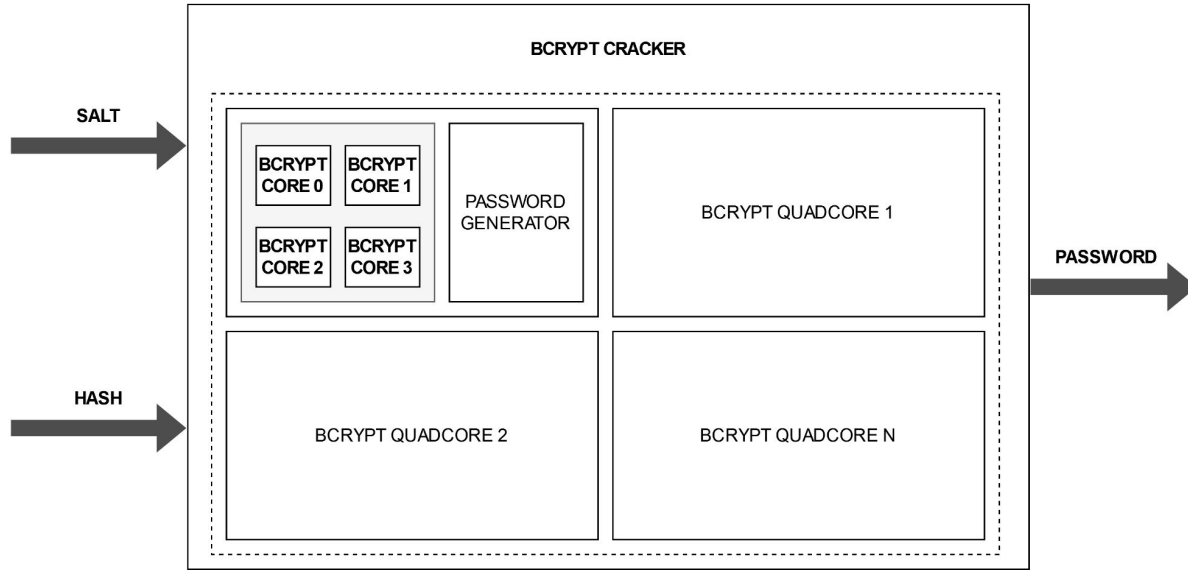
0

Forks

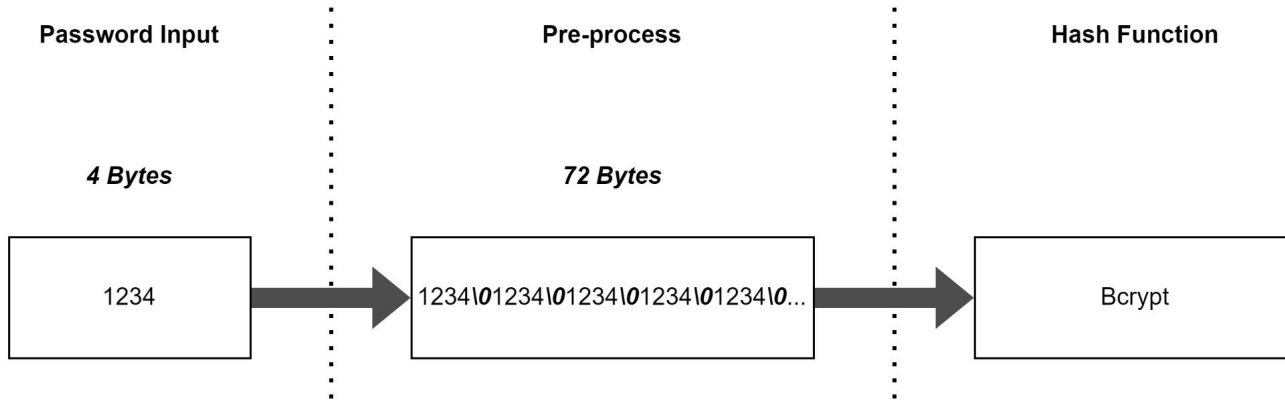


[https://github.com/rub-hgi/high-speed\\_bcrypt](https://github.com/rub-hgi/high-speed_bcrypt)

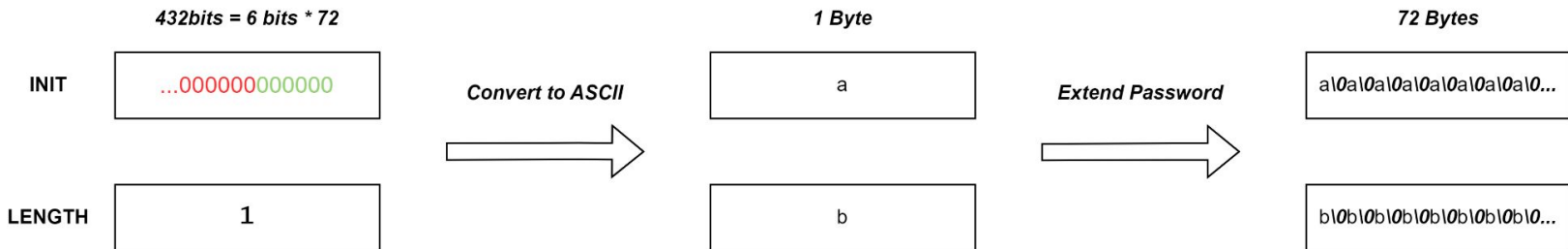
# Implémentation existante - Schéma



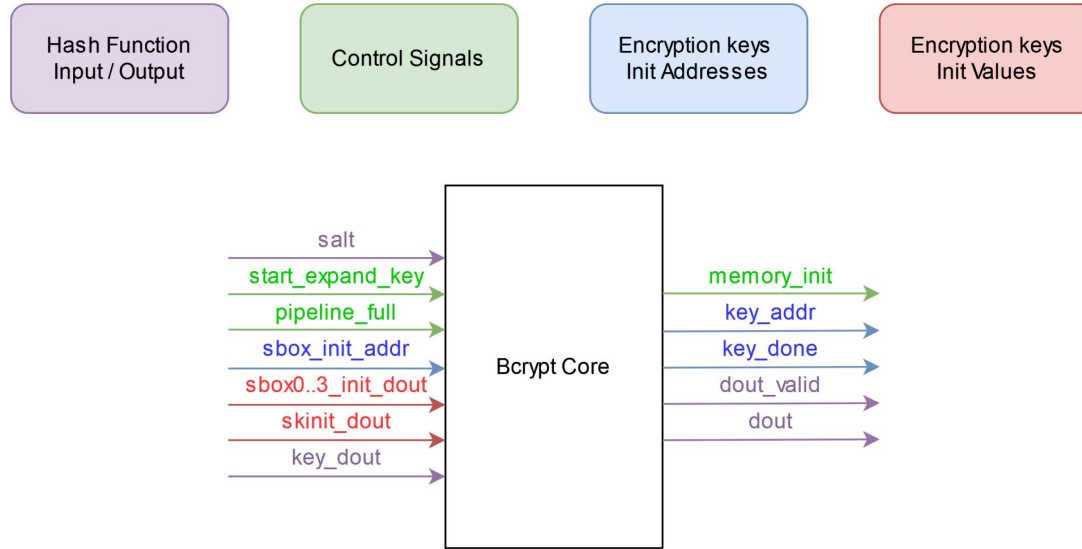
# Bcrypt - Password Hashing



Conversion Table				
0x00	0x01	0x02	0x1b	0x35
NULL '\0'	'a'	'b'	'A'	'0'



# Bcrypt Core Interface





# Implémentation existante - Problèmes

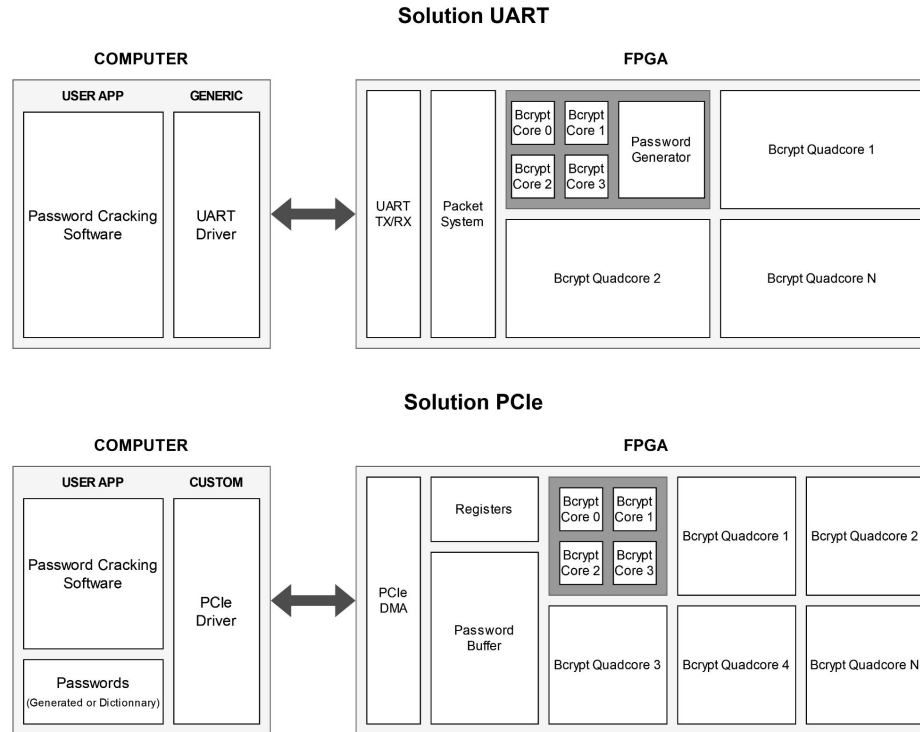


- Documentations
- Versions - Incohérences
- Testbenches incomplets
- Petites erreurs

---

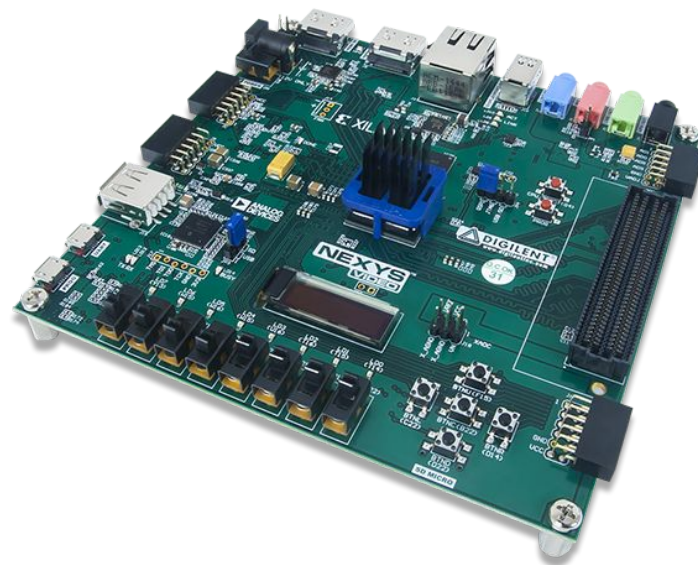
# Réalisations - Travail de bachelor

# Solutions - Schéma



# CARTE FPGA - Nexys Video

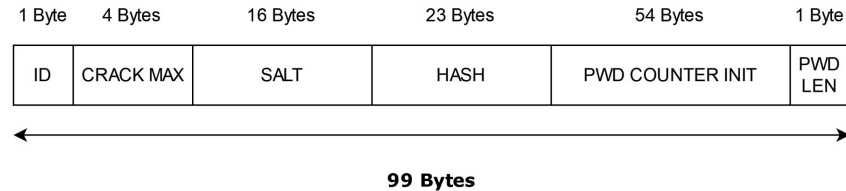
## Artix 7



# Solution UART - Initialisation des Quadcores

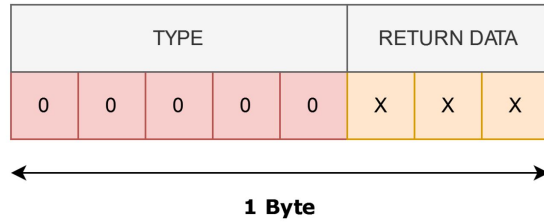
- Le nombre d'essais
- Le HASH et le SALT
- Initialisation du générateur de mots de passe

## PAYLOAD FORMAT - BCrypt QUADCORE INIT



# Solution UART - Réponse du système

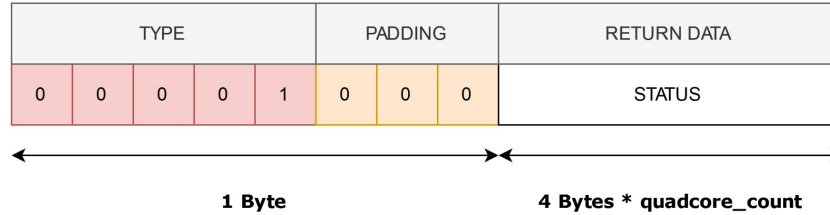
## PAYLOAD FORMAT - RETURN



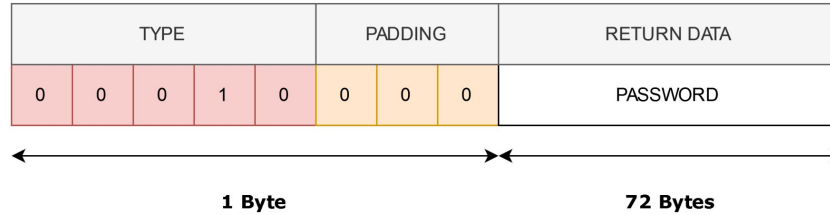
Return Code	Return
000	OK
001	Packet size greater than expected
010	Packet size smaller than expected
011	Quadcore ID not valid
100	CRC Error

# Solution UART - Retour du système

## PAYLOAD FORMAT - STATUS REPORT



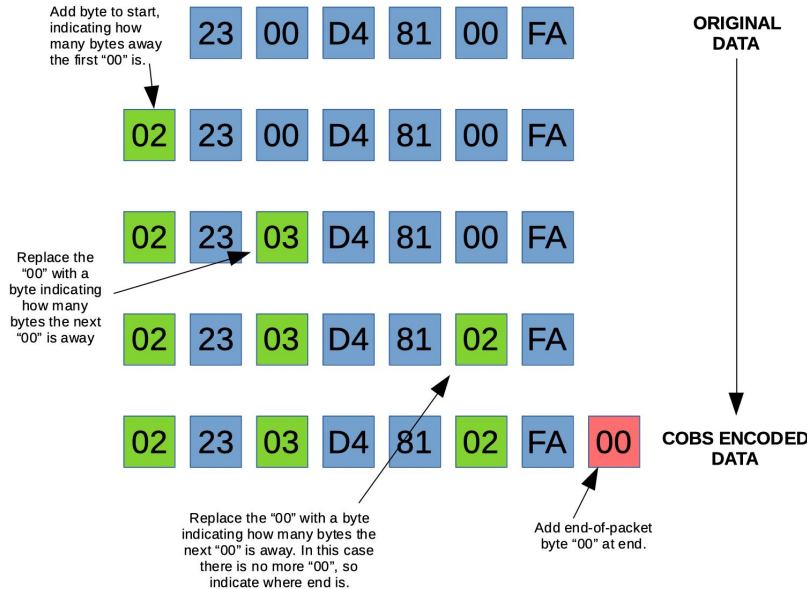
## PAYLOAD FORMAT - PASSWORD FOUND



# Solution UART - Encodage COBS



## THE COBS ENCODING PROCESS



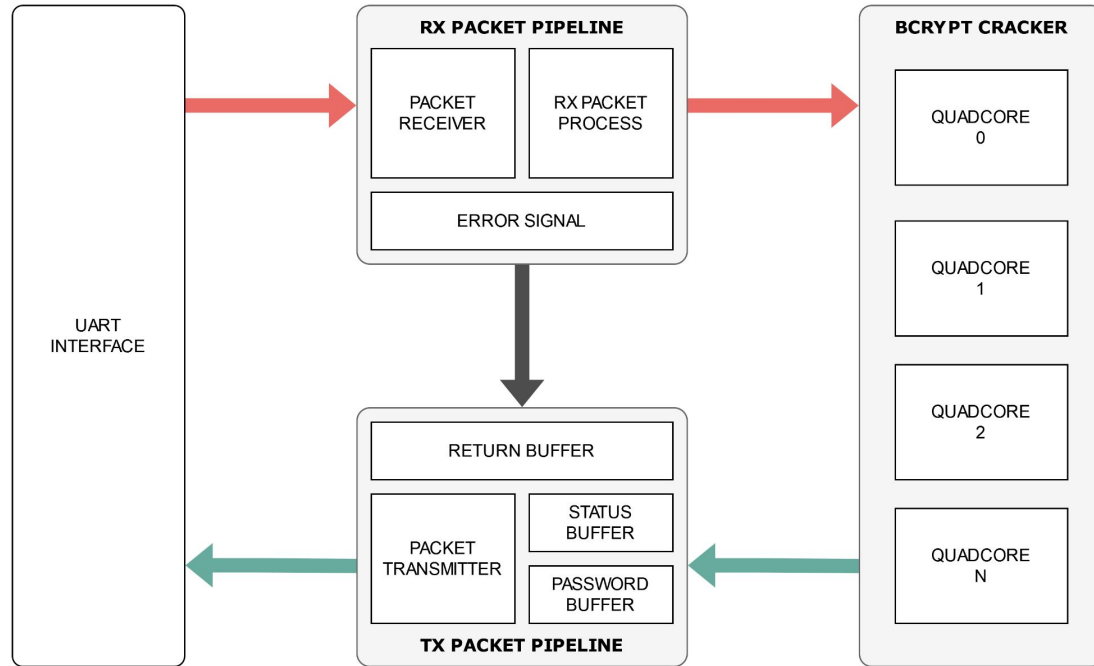
## PACKET FORMAT

1 Byte	1 Byte	Variable	1 Byte	1 Byte
COBS HEAD	PAYLOAD LENGTH	PAYLOAD	CRC	COBS END

<https://blog.mbedded.ninja/programming/serialization-for-mats/consistent-overhead-byte-stuffing-cobs/>

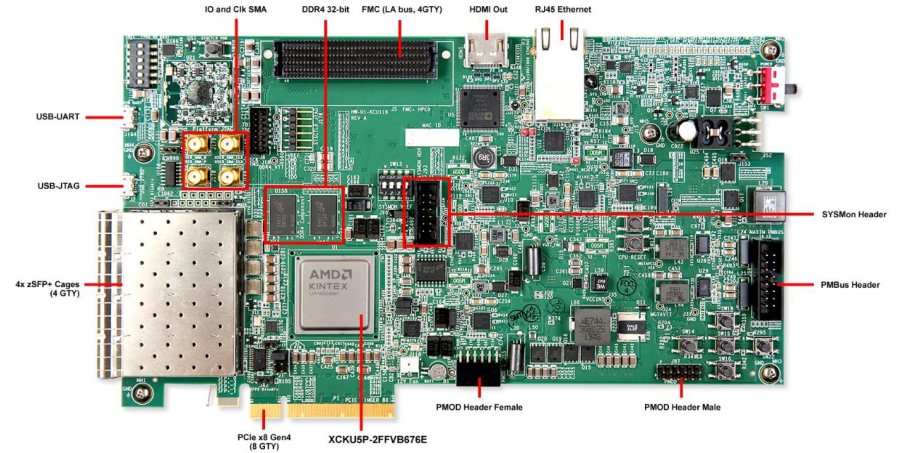


# Solution UART - Schéma

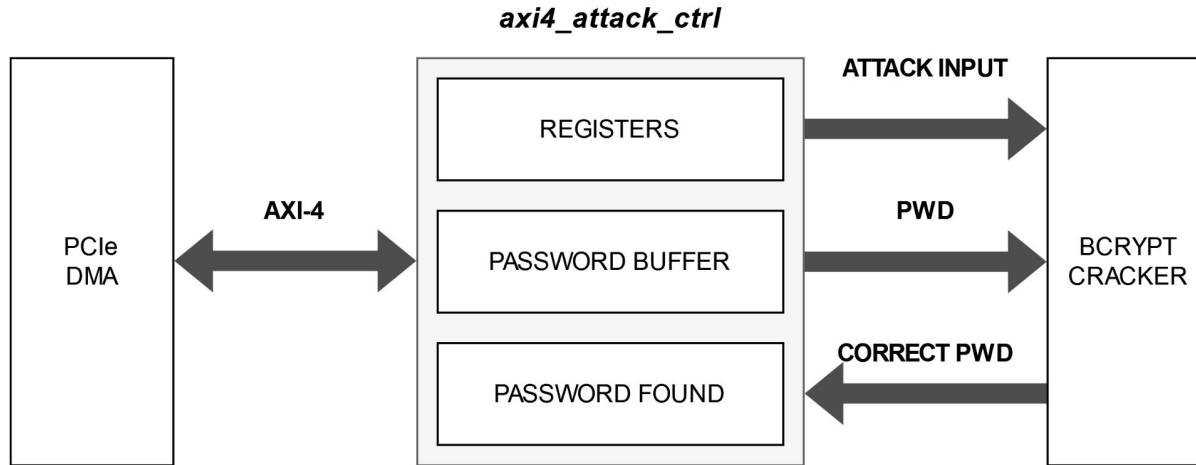


# Solution PCIe

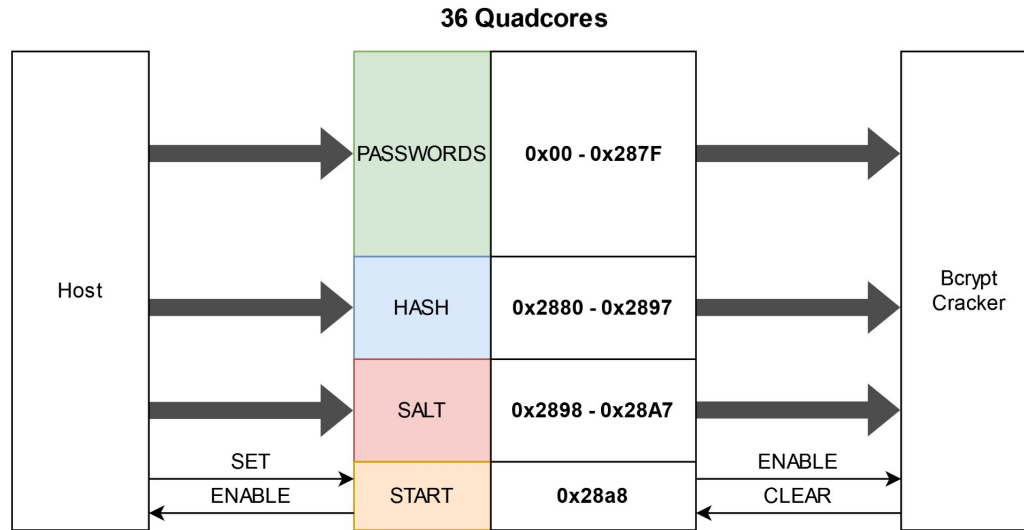
CARTE FPGA - KCU 116  
Kintex Ultrascale +



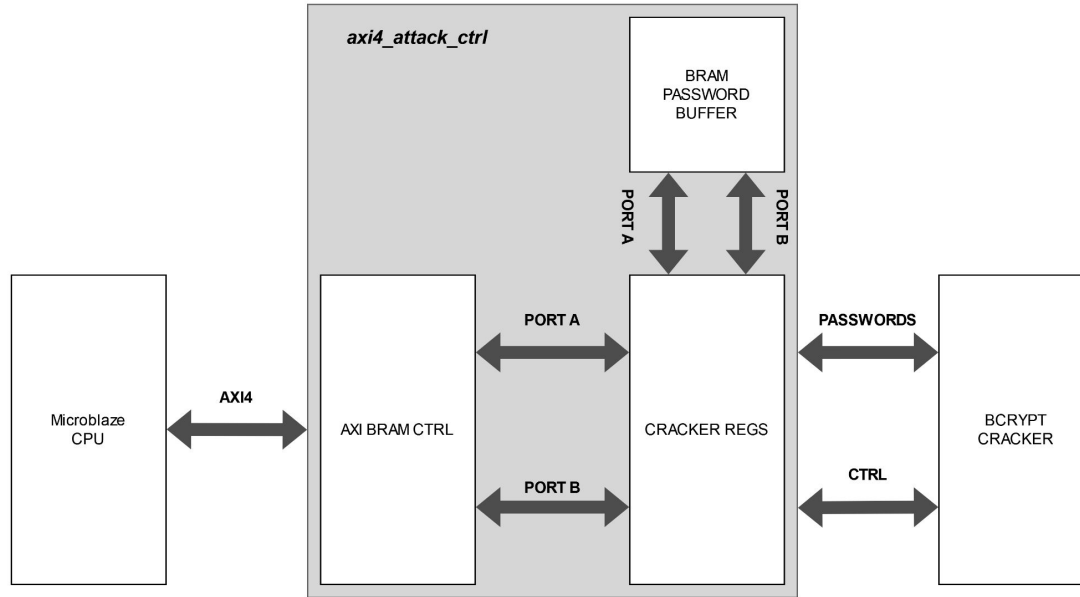
# Solution PCIe - Schéma Général



# Solution PCIe - Accès Mémoire



# Solution PCIe - Schéma de Test



---

# Résultats

# Résultats - Hashrate



Carte FPGA	Freq (MHz)	Quadcores Max.	Utilisations (%)	Hashrate (cost : 5)
<i>Nexys Video</i>	100	22	BRAM : 78.36, LUT : 75	<u>13'554 H/s</u>
<i>KCU 116</i>	100	36	BRAM : 97.50, LUT : 68	22'180 H/s
	200	36	BRAM : 97.50, LUT : 68	44'369 H/s
	250	36	BRAM : 97.50, LUT : 68	<u>55'450 H/s</u>
	275	< 30	-	< 50'820 H/s

# Résultats - Comparaisons

Architecture	Puissance (W)	Hashrate (cost : 5)	Efficacité Énergétique	
<i>FPGA (Nexys Video, 22 Quadcores, 100 MHz)</i>	4.38	13'554 H/s	3'094 H/J	11.140 MHash/Wh
<i>FPGA (KCU116, 36 Quadcores, 250 MHz)</i>	11.7	55'450 H/s	4'739 Hash/J	17.06 MHash/Wh
<i>CPU (AMD Ryzen 7 4800U, 16 threads)</i>	~25	8'200 H/s	328 H/J	1.18 MHash/Wh
<i>GPU (NVIDIA GTX 1660 Super)</i>	125	19'201 H/s	154 Hash/J	552.98 kHash/Wh



# Conclusion :



- Optimisation de l'implémentation Bcrypt
- Finir implémentation solution PCIe
- Mettre en place un driver linux pour PCIe

---

# Démo