

Introduction unité IFS



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License

Etienne Messerli, EMI ...

- Professeur depuis octobre 1995
 - Unité préparatoire de systèmes numériques: UPN
 - Unité d'architecture des ordinateurs ARO1 (1^{ère})
 - Unité de systèmes logiques SysLog1 et 2 (2^{ème})
 - Unité interface système à processeur IFS (2^{ème})
 - Unité de systèmes numériques CSN-E (3^{ème})
 - Unités à choix 3^{ème} année:
 - CSF (Conception de systèmes numériques sur FPGA)
 - SOCF (System on chip avec FPGA) **nouveau!**
 - Module master: ReCo (Reconfigurable Computing),
 - *Unité de théorie des circuits*

Formation, expérience professionnelle

- Diplômé de l'EPFL, Lausanne, 1986
- Diplômé de l'EIG, Genève, 1980
- Depuis 10.1995:
Professeur à l'EINEV, EIVD, puis HEIG-VD
- 10.1993-09.1995: Chargé de cours à l'EINEV
- 03.1987-09.1995: CYBELEC, Yverdon-les-Bains
- 07.1986-12.1986: Voyage Irlande, étude anglais
- 07.1980-06.1982: Brown-Boveri & Cie, Baden

Coordonnées EMI

- Etienne Messerli
- Acronyme: EMI
- Bureau: A09a, niveau A, aile est
- E-mail: etienne.messerli@heig-vd.ch
- Tél. 024/55 76 302

Activités

Répartition de mes activités :

- 60% enseignement
- 40% autres missions HES, soit :
 - Membre de l'institut REDS
 - Groupe: Conception intégrale de systèmes embarqués
 - Collaboration à la gestion de l'institut
 - Gestion et réalisation de projets Ra&D
 - Président du Conseil représentatif de l'école

Unité IFS

- Dotation cours et labo :

Semestre	Cours	Labo
S4, printemps	2	2

- Contrôle continu durant le semestre pour le cours et le labo
- pas d'examen d'unité

Relation étudiants → professeur

- Nommer un délégué de classe qui est le porte parole de l'ensemble des étudiant(e)s
- Son rôle est de transmettre, de manière anonyme, tous les messages de la classe au professeur
- Je communique ma réponse à la classe, si nécessaire j'ouvre une discussion

Déroulement du cours

Le cours est décomposé en 2 parties :

- ~ 50% présentation de la théorie et d'exemples.
- ~ 50% exercices (analyse, élaboration, correction, ...)
 - Correction par un étudiant ou le professeur au tableau noir
 - Un étudiant fait un corrigé et me le transmet
Je le valide, puis je le diffuse à la classe

Méthode de travail

- Prendre des notes
- Travailler régulièrement la matière
- Lire les supports de cours d'une séance à l'autre
- Faire les exercices d'une séance à l'autre
 - Nécessite un travail personnel
 - Permet de comprendre la matière au fur et à mesure
- **Objectif:**
poser des questions au cours suivant.

Support et documentation de cours

- Présentations PowerPoint
- Série d'exercices
- Documentation sur la carte REPTAR
 - Manuel de référence
 - Plan d'adressage Spartan 3 et 6
 - Datasheet module processeur DM3730
- Guide d'utilisation des infrastructures des labos A07-A09
- **Documentations:** <https://cyberlearn.hes-so.ch/>
Cours: 19_HEIG-VD_IFS, inscription: **ifs_2019**

Evaluation (contrôle continu)

- 3 Tests
 - durée ~60 minutes
 - les tests auront lieu : 3 avril, 8 mai, 5 juin
 - Documents autorisés :
 - Polycopiés et documentations fournies
 - Exercices et notes personnelles
 - Calculatrice
- Absence: note 1.0
 - si justificatif: travail de rattrapage (min un en fin d'unité)
- Moyenne du cours : moyenne calculée sur les N tests

Evaluation (contrôle continu)

- Evaluation du laboratoire:
 - 3 à 4 évaluations de travaux de laboratoire (écrit ou oral)
 - Laboratoire individuel ou par groupe de 2 étudiants
 - Moyenne : moyenne des évaluations, evtl avec pondération

Note finale de l'unité IFS

- Formule pour le calcul de la note finale de l'unité d'enseignement IFS :

$$\text{Note finale} = \text{cours} * 0.6 + \text{labo} * 0.4$$

- Contenu et calcul note finale:
 - Voir fiche d'unité: <https://gaps.heig-vd.ch/>

Planning IFS

- Voir fichier pdf

Contenu unité IFS ...

Fiche d'unité:
voir GAPS

• Cours:

- Introduction sur les systèmes embarqués
- Architecture d'un système à processeur (rappel)
- Plan d'adressage d'un système à processeur
- Interfaces simples entrées/sorties, gestion de flux
- Gestion par interruption, contrôleur d'interruption (vecteur)
- Comprendre et concevoir des mémoires pour système embarqués
- Notion d'abstraction du matériel (intro driver)
- Etude d'une chaîne d'acquisition classique comprenant un convertisseur A/D, D/A, codeur incrémental, ..

Contenu unité IFS ...

Fiche d'unité:
voir GAPS

• Laboratoire:

- Mise en œuvre E/S std FPGA-SP3 & CPU-ARM
- Interface E/S FPGA, inclus programme
- Réalisation d'un driver simple (abstraction du matériel)
- Interface avec gestion de flux et cohérence des données
- Réalisation d'une interface avec interruption
- Réalisation d'un système complet comprenant une partie matérielle, une interface et un logiciel

Objectifs unité IFS

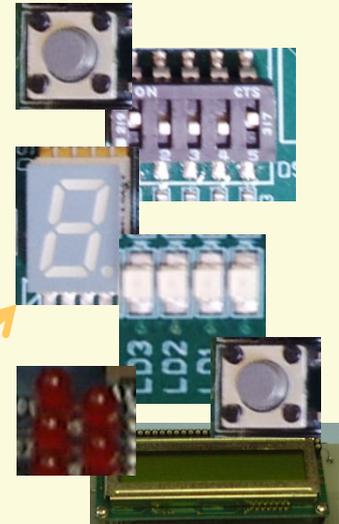
Software

```
#include <common.h>
typedef volatile unsigned short vushort;
//adresse pour les acces de SW4 du CPU
#define GPIO_DATAIN_REG_BK4
    *(volatile unsigned int *) (0x49054038)
#define MASK_SW4 0x40000000

int reptar_io_fpga(void)
{ unsigned short switches = 0;
/*Quitte l'application si SW4 du CPU active*/
while (!(GPIO_DATAIN_REG_BK4 & MASK_SW4)) {
    .....
}
return (0);
}
```

CPU

Input/Output

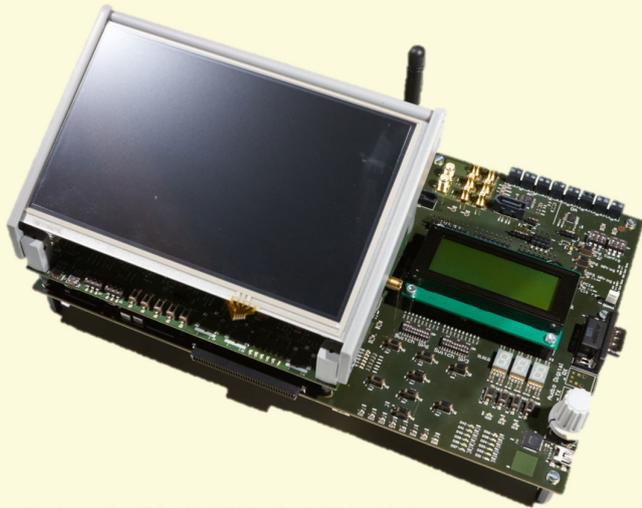


Logiciels et langage de description

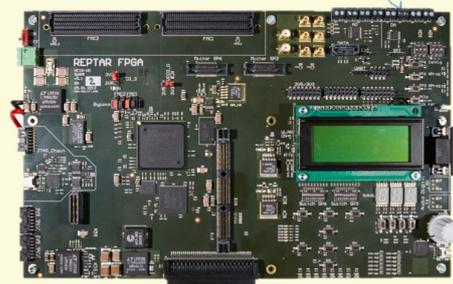
- Logisim et ISE Xilinx (placement-routage)
 - Logiciels permettant un *flow* complet pour réaliser un circuit logique programmable
 - Logisim: saisie graphique ou VHDL, simulation
 - ISE Xilinx: synthèse, intégration, programmation circuit
- Description des systèmes numériques sous forme schématique.
- Documentation
 - Guide d'utilisation de l'infrastructure des salles A07/A09
 - Manuel Logisim (idem ARO)

Plateforme embarquée pour labo

- Utilisation de la carte REPTAR
 - programmation du CPU-ARM (partie logiciel)
 - réalisation d'interface dans la FPGA Spartan-6



Carte CPU



Carte FPGA

E. Messerli (HES-SO / HEIG-VD / REDS), 2019

Introduction IFS, p 19 

Préparation pour

- Unités d'enseignements:
 - Architecture des systèmes à processeurs (ASP)
 - Conception de système numérique (CSN)
 - Programmation concurrente (PCO)
 - Programmation temps réel (PTR)
 - ...
- Applications avec microcontrôleurs (ARM)

Projet de semestre et diplôme !

E. Messerli (HES-SO / HEIG-VD / REDS), 2019

Introduction IFS, p 20 

Mes domaines : mots clés ...

Conception de systèmes numériques complexes

FPGA



```
Library IEEE;  
use IEEE.Std_Logic_1164.all;  
entity Flip_Flop is  
    port (D          : in Std_Logic;  
          Clock      : in Std_Logic;  
          Q          : out Std_Logic);  
end Flip_Flop;  
architecture Comport of Flip_Flop is  
begin  
    process (Clock)  
    begin  
        if Rising_Edge(Clock) then  
            Q <= D;  
        end if;  
    end process;  
end Comport;
```

VHDL

Outils EDA



Méthodologie

Vérification
VHDL-SystemVerilog

... domaines : mots clés ...

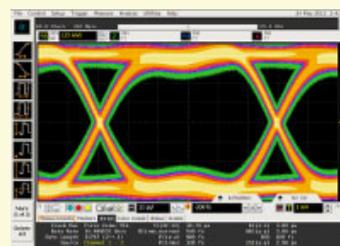


new FPGA Technology

Soft-Processor:



High-speed serial links :



Link at 14.1Giga

... domaines : mots clés

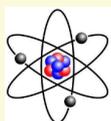
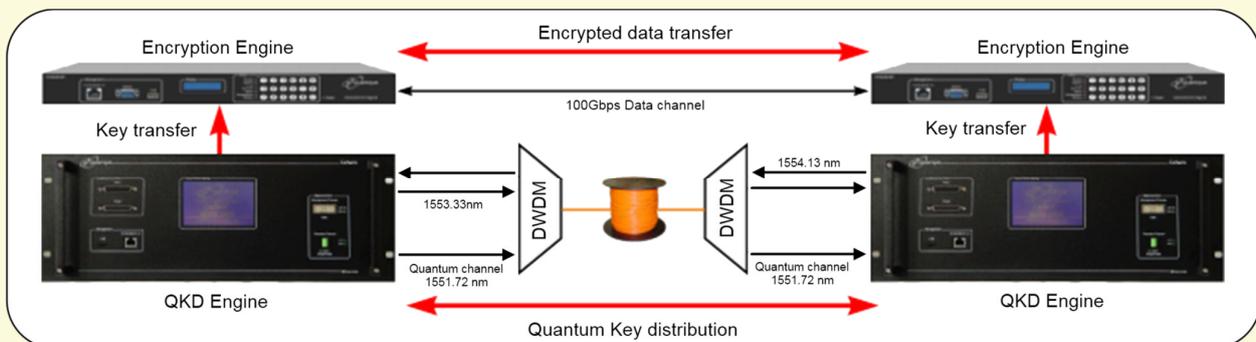
Révolution SoC-FPGA:
un seul circuit avec
 μ C + FPGA

High performance
Pipeline Co-Design



QCrypt: 100Gbit/s Encryption & Quantum Security

40..100Gb/s en**C**ryption and decryption
highest security for the next-generation communication speed

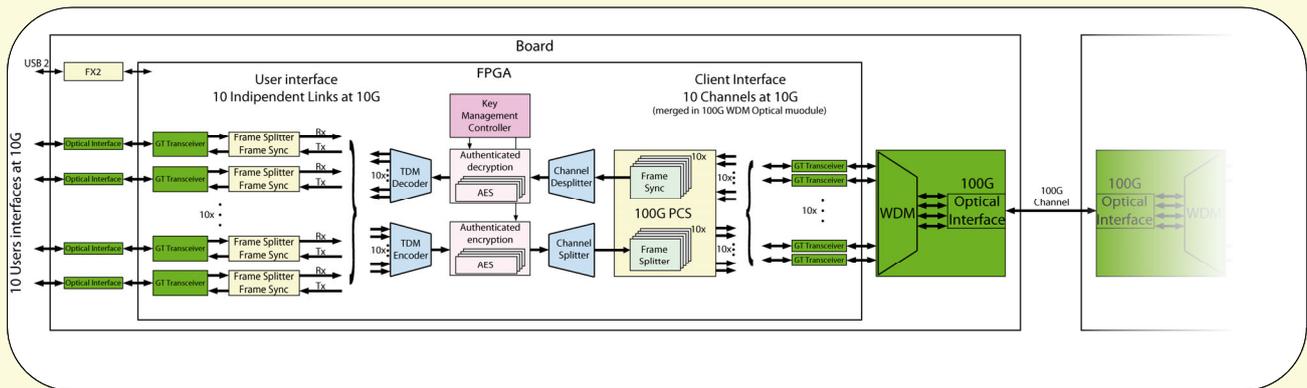


high-speed **Q**uantum Key Distribution
the impossibility of "Q cloning" is one of the best established facts in Science



Qcrypt System: Fast EnCryption

- User side: 10Giga Ethernet channels through 10 SPF+ optical modules
- Client side: 100Gbps channel using WDM optical module feeds with 10 high-speed serial links @ 10Giga
- All synchronization and channels splitting made into the FPGA

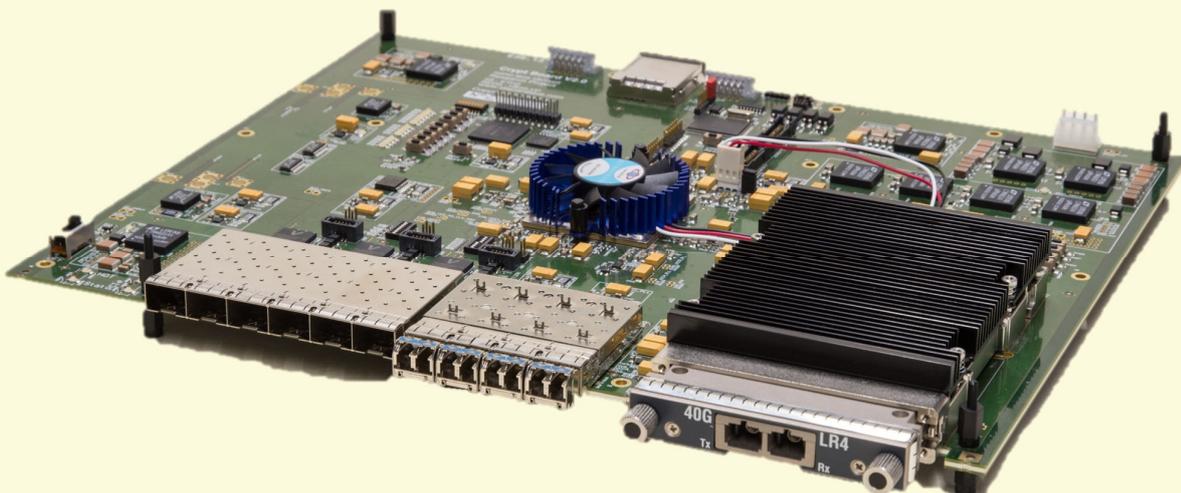


E. Messerli (HES-SO / HEIG-VD / REDS), 2019

Introduction IFS, p 25 **REDS**

Qcrypt System: Fast EnCryption

- PCB:** 26 layers, 52 high-speed serial links, 20 power supplies, backdrilled vias, routing FPGA with 2'000 pins
- FPGA main power supply:** 0,95V @ 40Amp



E. Messerli (HES-SO / HEIG-VD / REDS), 2019

QCRIPT – E. Messerli

Introduction IFS, p 26 **REDS**

Qcrypt System: Fast EnCryption

- Several clock domain, need FIFOs
 - 257.8125 MHz for transceiver
 - 160 MHz for PCS
- Variable data width
 - input 512 bits, scrambling 64, transceiver 40 bits
- BlockSync
 - to detect bloc (66 bits) alignment with control bits
 - need to analyse min 66 blocs !

