

# Table des matières

<b>Préface</b> . . . . .	1
Ahmed JERRAYA	
<b>Remerciements</b> . . . . .	3
Liliana ANDRADE et Frédéric ROUSSEAU	
<b>Présentation des auteurs</b> . . . . .	5
<b>Partie 1. MPSoC pour les télécoms</b> . . . . .	17
<b>Chapitre 1. Exigences matérielles pour les communications sans fil 6G</b> . . . . .	19
Stefan A. DAMJANCEVIC, Emil MATUS, Dmitry UTYANSKY, Pieter VAN DER WOLF et Gerhard P. FETTWEIS	
1.1. Introduction . . . . .	21
1.2. Étendue des charges de travail . . . . .	22
1.2.1. Vision, tendances et applications . . . . .	23
1.2.2. Spécifications normalisées . . . . .	25
1.2.3. Résultat des charges de travail . . . . .	30
1.3. Décomposition de l’algorithme GFDM . . . . .	32
1.3.1. Équation . . . . .	33
1.3.2. Graphe de traitement du flux de données et représentation matricielle . . . . .	33
1.3.3. Pseudo-code . . . . .	35
1.4. Exigences et considérations à l’égard de la précision de l’algorithme . .	36
1.5. Mise en œuvre . . . . .	40
1.5.1. Principes de mise en œuvre . . . . .	42

1.5.2. Exploration de l'espace de conception . . . . .	43
1.5.3. Mesures pour les cas d'utilisation bande basse et bande haute . . . . .	46
1.6. Conclusion . . . . .	48
1.7. Remerciements . . . . .	49
1.8. Bibliographie . . . . .	49

## **Chapitre 2. Vers un traitement en bande de base des communications sans fil de l'ordre du Tbit/s . . . . . 53**

Matthias HERRMANN et Norbert WEHN

2.1. Introduction . . . . .	54
2.2. Rôle de la microélectronique . . . . .	56
2.3. Vers les décodeurs à débit de 1 Tbit/s . . . . .	59
2.3.1. Turbo décodeur . . . . .	61
2.3.2. Décodeur LDPC . . . . .	62
2.3.3. Décodeur polaire . . . . .	63
2.4. Conclusion . . . . .	65
2.5. Remerciements . . . . .	65
2.6. Bibliographie . . . . .	65

## **Partie 2. Architectures MPSoC spécifiques à l'application . . . . . 69**

### **Chapitre 3. Automatisation pour l'industrie 4.0 avec des passerelles LoRaWan sécurisées . . . . . 71**

Marcello COPPOLA et George KORAROS

3.1. Introduction . . . . .	72
3.2. Sécurité de l'IIoT . . . . .	75
3.3. Sécurité LoRaWAN pour l'IIoT . . . . .	76
3.4. Modèle de menace . . . . .	78
3.4.1. Modèle d'attaque LoRaWAN . . . . .	78
3.4.2. Modèle d'attaque des nœuds de l'IIoT . . . . .	79
3.5. Chaîne de démarrage fiable avec STM32MP1 . . . . .	80
3.5.1. Base de confiance du nœud . . . . .	80
3.5.2. Micrologiciel de confiance dans STM32MP1 . . . . .	81
3.5.3. Environnement d'exécution fiable (TEE) et OP-TEE . . . . .	82
3.5.4. Considérations sur l'ordonnancement de l'OP-TEE . . . . .	84
3.5.5. Gestion de mémoire de l'OP-TEE . . . . .	84
3.5.6. API client de l'OP-TEE . . . . .	86
3.5.7. API du cœur interne TEE . . . . .	86
3.5.8. Racine et chaîne de confiance . . . . .	86
3.5.9. Clé matérielle unique . . . . .	87

3.5.10. Sécurité de l'horloge . . . . .	87
3.5.11. Opérations cryptographiques . . . . .	88
3.6. Passerelle LoRaWAN avec STM32MP1 . . . . .	88
3.7. Discussion et portée future . . . . .	90
3.8. Remerciements . . . . .	91
3.9. Bibliographie . . . . .	91

## **Chapitre 4. Accélération matérielle du stockage NVMe distribué et virtualisé . . . . . 95**

Julian CHESTERFIELD et Michail FLOURIS

4.1. Introduction . . . . .	96
4.1.1. Virtualisation et hyperviseurs traditionnels . . . . .	97
4.1.2. Architectures en nuage hyperconvergées <i>versus</i> désagrégées . . . . .	99
4.1.3. Stockage flash NVMe . . . . .	101
4.2. Motivation : le stockage NVMe pour le nuage . . . . .	101
4.2.1. Motivation pour un nouvel hyperviseur . . . . .	102
4.2.2. Motivation pour l'accélération du stockage désagrégé . . . . .	103
4.3. Modèle . . . . .	104
4.3.1. Optimisation des opérations d'E/S de l'hyperviseur . . . . .	104
4.3.2. Conception du stockage désagrégé accéléré . . . . .	107
4.4. Implémentation . . . . .	115
4.4.1. La plate-forme NexVisor . . . . .	115
4.4.2. Stockage désagrégé accéléré . . . . .	115
4.5. Résultats . . . . .	119
4.5.1. Lectures séquentielles . . . . .	119
4.5.2. Écritures séquentielles . . . . .	119
4.5.3. Lectures séquentielles sur un disque NVMe . . . . .	120
4.5.4. Performance du réseau . . . . .	122
4.6. Conclusion . . . . .	122
4.7. Bibliographie . . . . .	123

## **Chapitre 5. Plate-forme modulaire pour un environnement informatique de l'automobile du futur . . . . . 125**

Raphaël DAVID, Étienne HAMELIN, Paul DUBRULLE, Shuai LI,  
Philippe DORE, Alexis OLIVEREAU, Maroun OJAIL, Alexandre CARBON  
et Laurent LE GARFF

5.1. Introduction . . . . .	126
5.2. Aperçu de cette approche . . . . .	128
5.2.1. Calcul centralisé, données distribuées . . . . .	128
5.2.2. Modularité et hétérogénéité . . . . .	129
5.2.3. Outils de spécification, de configuration et d'intégration . . . . .	132

5.3. Résultats . . . . .	134
5.3.1. Plate-forme matérielle . . . . .	134
5.3.2. Architecture logicielle FACE . . . . .	141
5.3.3. Suite d'outils FACE . . . . .	146
5.4. Cas d'utilisation . . . . .	150
5.5. Conclusion . . . . .	153
5.6. Bibliographie . . . . .	154

**Chapitre 6. Architecture de serveur pour un centre de données post-Moore . . . . .**

Babak FALSAFI . . . . .	159
6.1. Introduction . . . . .	160
6.2. Contexte : les ordinateurs des années 1980 à l'origine des serveurs d'aujourd'hui . . . . .	162
6.3. Conception d'un serveur centré sur la mémoire . . . . .	164
6.4. Accélérateurs de gestion des données . . . . .	166
6.5. Contrôleurs de réseau intégrés . . . . .	168
6.6. Bibliographie . . . . .	169

**Partie 3. Exemples d'architectures et d'outils pour MPSoC . . . . .**

**Chapitre 7. SESAM, un environnement global pour le prototypage de systèmes cyberphysiques . . . . .**

Amir CHARIF, Arief WICAKSANA, Salah-Eddine SAIDI, Tanguy SASSOLAS, Caaliph ANDRIAMISAINA et Nicolas VENTROUX . . . . .	175
7.1. Introduction . . . . .	176
7.2. Un aperçu de la plate-forme SESAM . . . . .	177
7.2.1. Prototypage de systèmes multi-abstraction . . . . .	177
7.2.2. Évaluation des propriétés extrafonctionnelles du système . . . . .	178
7.3. VPSim : prototypage virtuel rapide et facile . . . . .	179
7.3.1. Codage de périphériques en Python . . . . .	179
7.3.2. Interface <i>ModelProvider</i> . . . . .	181
7.3.3. Support QEMU . . . . .	183
7.3.4. Monitoring de la simulation en ligne . . . . .	185
7.3.5. Méthodes d'accélération . . . . .	186
7.4. Prototypage hybride . . . . .	186
7.4.1. Mode de cosimulation . . . . .	188
7.4.2. Mode de coémulation . . . . .	189
7.4.3. Analyse des performances d'exécution et fonctionnalités de débogage . . . . .	189

7.5. Le FMI pour la cosimulation . . . . .	191
7.5.1. Interface FMI ( <i>Functional Mock-up Interface</i> ) . . . . .	191
7.5.2. Intégration de VPSim dans la cosimulation FMI . . . . .	192
7.6. Conclusion . . . . .	196
7.7. Bibliographie . . . . .	196

## **Chapitre 8. StaccatoLab, un modèle de calcul des flots de données à grande échelle . . . . . 199**

Kees VAN BERKEL

8.1. Introduction . . . . .	200
8.2. Flot de données statique . . . . .	204
8.2.1. Flot de données synchrone . . . . .	205
8.2.2. Flot de données cyclostatique . . . . .	209
8.2.3. Transformations de diagrammes de flot de données . . . . .	211
8.3. Flot de données dynamique . . . . .	211
8.3.1. Flot de données dépendant des données . . . . .	212
8.3.2. Flot de données non déterminé . . . . .	215
8.4. Modèles d'exécution de flot de données . . . . .	219
8.4.1. Bref aperçu de la théorie de flot de données . . . . .	220
8.4.2. Modèle d'exécution de StaccatoLab . . . . .	222
8.5. StaccatoLab . . . . .	225
8.5.1. Description et analyse de diagrammes de flot de données . . . . .	225
8.5.2. Synthèse Verilog . . . . .	226
8.6. Calcul des flots de données à grande échelle ? . . . . .	228
8.6.1. Quel type d'applications ? . . . . .	228
8.6.2. Efficacité de l'approche ? . . . . .	229
8.6.3. Rendement de l'approche ? . . . . .	230
8.7. Remerciements . . . . .	231
8.8. Bibliographie . . . . .	231

## **Chapitre 9. Caméras intelligentes et MPSoC . . . . . 235**

Marilyn WOLF

9.1. Introduction . . . . .	235
9.2. Premiers processeurs vidéo VLSI . . . . .	237
9.3. Processeurs de signal vidéo . . . . .	238
9.4. Accélérateurs . . . . .	239
9.5. Du VSP au MPSoC . . . . .	242
9.6. Unités de traitement graphique . . . . .	244
9.7. Réseaux de neurones et unités de traitement tensoriel . . . . .	244
9.8. Conclusion . . . . .	246
9.9. Bibliographie . . . . .	247

<b>Chapitre 10. Compilation et optimisation pour plates-formes multicœurs hétérogènes</b> . . . . .	251
Weihua SHENG, Jeronimo CASTRILLON et Rainer LEUPERS	
10.1. Introduction . . . . .	252
10.2. Modélisation du flot de données . . . . .	256
10.2.1. Concepts généraux . . . . .	256
10.2.2. Réseaux de processus . . . . .	257
10.2.3. C pour les réseaux de processus . . . . .	258
10.3. Infrastructure de compilation de type source à source . . . . .	263
10.3.1. Logique de conception . . . . .	263
10.3.2. Stratégie de mise en œuvre . . . . .	265
10.4. Distribution logicielle . . . . .	268
10.4.1. Analyse KPN . . . . .	269
10.4.2. Mappage statique du KPN . . . . .	270
10.4.3. Mappage hybride du KPN . . . . .	272
10.5. Résultats . . . . .	272
10.5.1. Applications et expériences . . . . .	273
10.5.2. Redéfinition de la cible . . . . .	281
10.6. Conclusion . . . . .	282
10.7. Bibliographie . . . . .	284
<b>Liste des auteurs</b> . . . . .	289
<b>Index</b> . . . . .	293
<b>Sommaire de <i>Systèmes multiprocesseurs sur puce 1</i></b> . . . . .	295