

# 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. Les processeurs</b> . . . . .	13
<b>Chapitre 1. Processeurs pour l'Internet des objets</b> . . . . .	15
Pieter VAN DER WOLF et Yankin TANURHAN	
1.1. Introduction . . . . .	16
1.2. Processeurs polyvalents pour les dispositifs IoT Edge à faible consommation . . . . .	17
1.2.1. Traitement de contrôle, DSP et apprentissage automatique . . . . .	17
1.2.2. Configurabilité et extensibilité . . . . .	19
1.3. Inférence d'apprentissage automatique . . . . .	21
1.3.1. Conditions requises pour l'inférence d'apprentissage automatique basse/moyenne gamme . . . . .	23
1.3.2. Capacités du processeur pour l'inférence d'apprentissage automatique à faible consommation . . . . .	29
1.3.3. Bibliothèque logicielle pour l'inférence d'apprentissage automatique . . . . .	32
1.3.4. Exemples d'applications et de benchmarks d'apprentissage automatique . . . . .	36

1.4. Conclusion . . . . .	40
1.5. Bibliographie . . . . .	40

## **Chapitre 2. Une approche qualitative de l'architecture pluricœur . . . . .** 43

Benoît DUPONT DE DINECHIN

2.1. Introduction . . . . .	44
2.2. Motivations et contexte . . . . .	45
2.2.1. Processeurs multicœurs et pluricœurs . . . . .	45
2.2.2. Inférence d'apprentissage automatique . . . . .	47
2.2.3. Exigences des applications . . . . .	49
2.3. Processeur pluricœur MPPA3 . . . . .	52
2.3.1. Architecture globale . . . . .	52
2.3.2. Complexe de calcul . . . . .	55
2.3.3. Cœur VLIW . . . . .	56
2.3.4. Coprocesseur tensoriel . . . . .	59
2.4. Environnements logiciels MPPA3 . . . . .	61
2.4.1. Calcul haute performance . . . . .	61
2.4.2. Générateur de code KaNN . . . . .	63
2.4.3. Calcul haute intégrité . . . . .	66
2.5. Conclusion . . . . .	67
2.6. Bibliographie . . . . .	68

## **Chapitre 3. Plural, pluricœur haute performance à faible consommation d'énergie . . . . .** 73

Ran GINOSAR

3.1. Introduction . . . . .	74
3.2. Travaux connexes . . . . .	75
3.3. Architecture pluricœur Plural . . . . .	75
3.4. Modèle de programmation Plural . . . . .	76
3.5. Ordonnanceur/synchroniseur matériel Plural . . . . .	79
3.6. Réseaux sur puce Plural . . . . .	82
3.6.1. Ordonnanceur NoC . . . . .	82
3.6.2. Mémoire partagée NoC . . . . .	83
3.7. Accélérateurs matériels et logiciels pour l'architecture Plural . . . . .	83
3.8. Logiciel système Plural . . . . .	84
3.9. Outils Plural pour le développement logiciel . . . . .	86
3.10. Algorithme de multiplication matricielle sur l'architecture Plural . . . . .	87
3.11. Conclusion . . . . .	89
3.12. Bibliographie . . . . .	90

<b>Chapitre 4. Systèmes multiprocesseurs basés sur un ASIP pour l'efficacité des CNN</b> . . . . .	93
Andreas BYTYN, René AHLSDORF et Gerd ASCHEID	
4.1. Introduction . . . . .	94
4.2. Travaux connexes . . . . .	95
4.3. Architecture ASIP . . . . .	98
4.4. Mise à l'échelle monocœur . . . . .	100
4.5. Présentation du système sur puce multiprocesseur (MPSoC) . . . . .	103
4.6. Exploration des paramètres du NoC . . . . .	105
4.7. Conclusion . . . . .	108
4.8. Bibliographie . . . . .	109
<b>Partie 2. La mémoire</b> . . . . .	113
<b>Chapitre 5. Relever le défi de la localité des données dans les MPSoC</b> . . . . .	115
Sven RHEINDT, Akshay SRIVATSA, Oliver LENKE, Lars NOLTE, Thomas WILD et Andreas HERKERSDORF	
5.1. Motivation . . . . .	116
5.2. Plate-forme cible MPSoC . . . . .	119
5.3. Travaux connexes . . . . .	120
5.4. Cohérence à la demande : cohérence de cache par région . . . . .	121
5.4.1. RBCC <i>versus</i> cohérence globale . . . . .	122
5.4.2. Extensions du système d'exploitation pour la cohérence à la demande . . . . .	123
5.4.3. Gestionnaire de région de cohérence . . . . .	124
5.4.4. Évaluations expérimentales . . . . .	128
5.4.5. RBCC et placement des données . . . . .	131
5.5. Accélération proche-mémoire . . . . .	132
5.5.1. Accélérateur de synchronisation proche-mémoire . . . . .	134
5.5.2. Accélérateur de gestion des files d'attente proche-mémoire . . . . .	137
5.5.3. Accélérateur de copie de graphes proche-mémoire . . . . .	141
5.5.4. Accélérateur proche-cache . . . . .	145
5.6. Vue d'ensemble . . . . .	147
5.7. Conclusion . . . . .	149
5.8. Remerciements . . . . .	149
5.9. Bibliographie . . . . .	150

**Chapitre 6. mMPU, une architecture polyvalente de calcul dans la mémoire basée sur memristor . . . . . 155**

Adi ELIAHU, Rotem BEN HUR, Ameer HAJ ALI et Shahar KVATINSKY

- 6.1. Introduction. . . . . 156
- 6.2. Porte MAGIC NOR . . . . . 157
- 6.3. Algorithmes en mémoire pour la réduction de la latence . . . . . 159
- 6.4. Méthodes de synthèse et de placement en mémoire . . . . . 160
  - 6.4.1. SIMPLE . . . . . 160
  - 6.4.2. SIMPLER . . . . . 163
- 6.5. Conception du contrôleur de mémoire . . . . . 164
- 6.6. Conclusion . . . . . 167
- 6.7. Bibliographie. . . . . 167

**Chapitre 7. Élimination des appels externes des accès mémoire dans la traduction binaire dynamique. . . . . 171**

Antoine FARAVELON, Olivier GRUBER et Frédéric PÉTROU

- 7.1. Introduction. . . . . 172
- 7.2. Émulation des accès mémoire . . . . . 175
- 7.3. Conception de notre solution . . . . . 180
- 7.4. Mise en œuvre . . . . . 184
  - 7.4.1. Module noyau . . . . . 185
  - 7.4.2. Traduction binaire dynamique . . . . . 186
  - 7.4.3. Optimisation de notre chemin lent . . . . . 188
- 7.5. Évaluations expérimentales . . . . . 191
  - 7.5.1. Analyse de la performance de l’émulation . . . . . 192
  - 7.5.2. Performance de notre solution . . . . . 193
  - 7.5.3. Optimisation du chemin lent . . . . . 196
- 7.6. Travaux connexes . . . . . 197
- 7.7. Conclusion . . . . . 200
- 7.8. Bibliographie. . . . . 202

**Chapitre 8. Méthodes matérielles de distribution des accès en banc mémoire. . . . . 205**

Arthur VIANES et Frédéric ROUSSEAU

- 8.1. Introduction. . . . . 206
  - 8.1.1. Contexte . . . . . 206
  - 8.1.2. Architecture MPSoC . . . . . 207
  - 8.1.3. Composant d’interconnexion . . . . . 208

8.2. Fondements de la mémoire découpée en bancs . . . . .	209
8.2.1. Mémoire découpée en bancs . . . . .	209
8.2.2. Conflit en bancs mémoire et granularité . . . . .	210
8.2.3. Utilisation efficace des bancs mémoire : entrelacement . . . . .	212
8.3. Résumé des approches logicielles. . . . .	215
8.3.1. Méthode par ajout de données inutiles ( <i>padding</i> ) . . . . .	215
8.3.2. Ordonnancement statique des accès mémoire . . . . .	216
8.3.3. Besoin d’approches matérielles. . . . .	217
8.4. Approches matérielles . . . . .	217
8.4.1. Système d’indexation en modulo de nombre premier. . . . .	217
8.4.2. Modes d’entrelacement utilisant des fonctions de hachage . . . . .	219
8.5. Modélisation et expérimentation . . . . .	228
8.5.1. Mise en œuvre du simulateur . . . . .	228
8.5.2. Mise en œuvre de l’interconnexion du cluster MPPA de Kalray . . . . .	229
8.5.3. Objectifs et méthode . . . . .	230
8.5.4. Résultats et discussions . . . . .	231
8.6. Conclusion . . . . .	239
8.7. Bibliographie. . . . .	239

### **Partie 3. L’interconnexion et les interfaces . . . . . 241**

#### **Chapitre 9. NoC, la technologie de communication des MPSoC . . . . . 243**

K. Charles JANAC

9.1. Histoire : transition des bus et des <i>crossbars</i> vers les NoC . . . . .	244
9.1.1. Architecture NoC . . . . .	248
9.1.2. Extension de la comparaison de bus aux <i>crossbars</i> . . . . .	254
9.1.3. Résumé et conclusion de la comparaison bus, <i>crossbar</i> et NoC. . . . .	255
9.2. Configurabilité NoC. . . . .	255
9.2.1. Flux de conception guidé par l’homme . . . . .	256
9.2.2. Sensibilisation au placement physique et conception de l’architecture NoC . . . . .	257
9.3. Services au niveau du système . . . . .	260
9.3.1. Qualité de service (QoS) et arbitrage . . . . .	260
9.3.2. Débogage matériel et analyse des performances. . . . .	261
9.3.3. Sûreté et sécurité fonctionnelles . . . . .	261
9.4. Cohérence du cache matériel . . . . .	265
9.4.1. Protocoles, sémantique et messagerie NoC. . . . .	266

9.5. Développements futurs de la technologie NoC . . . . . 267

    9.5.1. Synthèse de la topologie et connaissance du plan d'étage . . . . . 267

    9.5.2. Résilience avancée et sécurité fonctionnelle  
pour les véhicules autonomes. . . . . 268

    9.5.3. Alternatives aux architectures von Neumann pour les SoC . . . . . 268

    9.5.4. *Chiplets* et connectivité NoC *multi-die* . . . . . 272

    9.5.5. Automatisation du logiciel d'exécution . . . . . 273

    9.5.6. Instrumentation, diagnostics et analyses pour la performance,  
la sûreté et la sécurité . . . . . 274

9.6. Conclusion . . . . . 274

9.7. Bibliographie. . . . . 275

**Chapitre 10. Calcul de l'énergie minimale par ajustement  
des tensions d'alimentation et de seuil. . . . . 277**

Jun SHIOMI et Tohru ISHIHARA

10.1. Introduction . . . . . 278

10.2. Mémoire à base de cellules standard pour le calcul  
de l'énergie minimale . . . . . 280

    10.2.1. Vue d'ensemble des mémoires sur puce basse tension . . . . . 280

    10.2.2. Stratégie de conception pour des SCM efficaces  
en termes de surface et d'énergie . . . . . 285

    10.2.3. Conception de mémoires hybrides pour des systèmes  
de mémoire efficaces en termes d'énergie et de surface . . . . . 287

    10.2.4. Polarisation de corps comme alternative au *power gating* . . . . . 288

10.3. Suivi du point d'énergie minimale . . . . . 289

    10.3.1. Théorie de base . . . . . 290

    10.3.2. Algorithmes et implémentation . . . . . 296

    10.3.3. Approche basée sur les systèmes d'exploitation pour le suivi  
du point d'énergie minimale . . . . . 299

10.4. Conclusion . . . . . 302

10.5. Remerciements . . . . . 303

10.6. Bibliographie . . . . . 303

**Chapitre 11. Cohérence des communications lors de la migration  
de tâches matérielles . . . . . 309**

Arief WICAKSANA, Olivier MULLER, Frédéric ROUSSEAU et Arif SASONGKO

11.1. Introduction . . . . . 310

    11.1.1. Architectures reconfigurables . . . . . 310

    11.1.2. Contribution . . . . . 311

---

11.2. Contexte . . . . .	312
11.2.1. Définitions . . . . .	312
11.2.2. Scénario du problème et défis techniques . . . . .	313
11.3. Travaux connexes . . . . .	315
11.3.1. Changement de contexte matériel. . . . .	315
11.3.2. Gestion de la communication . . . . .	317
11.4. Méthodologie de communication proposée pour le changement de contexte matériel . . . . .	318
11.5. Implantation de la gestion de la communication sur des architectures reconfigurables . . . . .	321
11.5.1. Canaux reconfigurables en FIFO . . . . .	322
11.5.2. Infrastructure de communication . . . . .	323
11.6. Résultats expérimentaux . . . . .	326
11.6.1. Configuration . . . . .	326
11.6.2. Scénario d'expérimentation . . . . .	327
11.6.3. Surcoûts en ressources . . . . .	328
11.6.4. Impact sur le temps total d'exécution . . . . .	330
11.6.5. Impact sur le temps d'extraction et de restauration du contexte . . . . .	332
11.6.6. Réactivité du système aux demandes de changement de contexte. . . . .	333
11.6.7. Migration des tâches matérielles entre FPGA hétérogènes . . . . .	337
11.7. Conclusion . . . . .	340
11.8. Bibliographie . . . . .	341
<b>Liste des auteurs . . . . .</b>	<b>345</b>
<b>Index . . . . .</b>	<b>349</b>
<b>Sommaire de <i>Systèmes multiprocesseurs sur puce 2</i>. . . . .</b>	<b>351</b>