Qtechnology met de l'APU dans sa vision

Les unités de traitement accélérées font leur entrée dans les dernières générations de caméras intelligentes du fabricant danois.

Qtechnology a sauté le pas ! Sa dernière génération de caméra intelligente utilise les unités de traitement accélérées (APU) de la série G embarquée d'AMD pour tous les calculs de haut niveau, et une architecture FPGA à faible coût pour l'interface de capteur et les prétraitements d'images. Pourquoi ce changement du FPGA pour des puces de type x86 ? Pour réduire les coûts et les temps de développement et pour gagner en polyvalence, tout en préservant la performance. « *Dans les configurations basées sur des FPGA avec cœur CPU dur, la plupart des applications de vision exigent que les algorithmes de traitement d'images les plus intensifs soient implémentés dans la matrice FPGA. Cela est très contraignant car il faut, pour chaque application, dédier un cycle de conception séparé à l'écriture du code en langage de développement VHDL (VHSIC hardware description langage) et à celle des drivers associés. Une simple modification de l'algorithme, du flot de données ou même du processus de débogage et de vérification impose une longue itération d'implémentation du FPGA* », explique Ricardo Ribalda, ingénieur firmware.

La caméra intelligente QT-XV70 inaugure l'utilisation des APU chez Qtechnology.

Simples et… ouverts

En optant pour des unités de traitement accélérées, les ingénieurs danois ont au contraire décomplexifié et accéléré leurs développements. « *Les APU x86, T56N de la série G embarquée d'AMD nous ont permis d'atteindre une efficacité de développement précédemment inenvisageable, en grande partie en puisant dans des bibliothèques de traitement d'images open source et de tierces-parties comme OpenCV, Mathworks Matlab et Halcon* », explique Ricardo Ribalda. Les phases de test sont, elles aussi, plus simples qu'auparavant. « *Le x86 peut être testé avec des outils standard incluant des analyseurs de latence et de performance (perf, ftrace), des débogueurs modernes et des environnements de développement intégrés (Eclipse, gdb), des analyseurs de puissance (powertop)… Tous ces outils sont déployés facilement avec des outils fournis par des projets de collaboration open source comme le Yocto Project, qui offre aux concepteurs des gabarits, des outils et des méthodes qui aident à créer des systèmes Linux personnalisés pour systèmes embarqués, indépendamment de l'architecture matérielle. Les applications peuvent être compilées et testées sur n'importe quel ordinateur, en se passant de compilateur croisé spécifique ou d'un BSP (Board Support Package) à support réduit* », poursuit l'ingénieur firmware.

De nombreux avantages

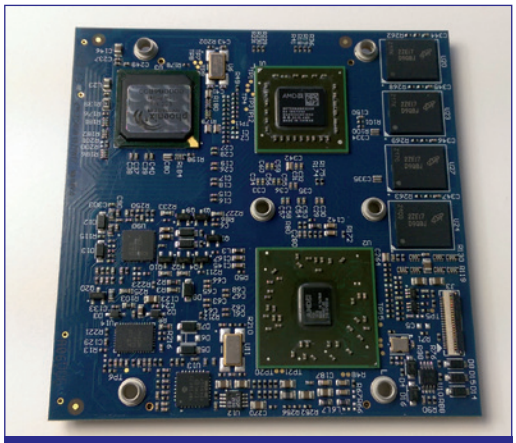
Sur une puce APU, la cohabitation d'un GPU avec le CPU permet de décharger ce dernier en confiant au GPU les traitements intensifs de pixels de l'application de vision. Libéré de ces traitements, le CPU peut répondre à d'autres interruptions avec une latence beaucoup plus faible, améliorant considérablement la performance temps réel du système global. Et « *avec cette architecture, toute la plate-forme est gérable par un noyau Linux standard, qui ne demande qu'un support modéré à chaque révision* », note Ricardo Ribalda.

« *Pour simplifier encore le développement et comprimer les cycles, nous avons intégré l'APU dans un Panel PC industriel et dans un module de contrôle d'E/S de 80 mm x 80 mm, nous réservant la possibilité de réutiliser du code dans différents systèmes via une seule distribution basée sur Yocto* », note l'ingénieur firmware. L'ouverture va encore plus loin. En effet, l'emploi de standards industriels comme Ethercat et Modbus passe « *simplement* » par l'intégration de bibliothèques existantes, généralement déjà développées sur x86. Et pour les applications nécessitant des moyens de calculs plus importants, la plate-forme APU sélectionnée par Qtechnology permet d'améliorer la performance du système en couplant

l'APU avec un GPU Radeon d'AMD en format MXM. Mais surtout, « *le choix d'un APU au lieu d'un FPGA comme calculateur central de nos caméras intelligentes nous a permis aussi de mieux profiter d'outils de développement ouverts tels qu'OpenCL. Cette plate-forme de programmation non propriétaire permet d'accroître la performance des traitements en parallèle, tout en développant une seule base de code source pour une large gamme de systèmes de vision* », note Glöde Madsen. Cela permet aussi d'offrir des fonctionnalités poussées si nécessaire. Enfin, le support de x86 facilite l'intégration avec les réseaux et des systèmes de contrôle distribués apportant des bénéfices additionnels aux applications hébergées. « *Il nous a été facile d'implémenter des fonctions de télégestion et de débogage via Internet, ce qui a amélioré notre service client et nous a permis de fournir des données temps réel à nos clients à des fins de « fouilles de données »* », poursuit le confondateur de Qtechnology.

Le FPGA ne disparaît pas

Le fabricant de caméras n'a pas totalement abandonné les FPGA. « *Nous avons seulement échangé notre architecture traditionnelle purement FPGA contre une approche APU + FPGA faible coût ; nous avons partitionné le système de manière à ce que le FPGA économique soit dédié uniquement à l'interfaçage du capteur et à un flux de tâches de vision fixe, consistant en prétraitements d'images. Cette fonction en FPGA dispose comme interface de Video4Linux2 (V4L2), l'environnement API + driver standard de Linux pour périphériques de capture et sortie vidéo. Toutes les applications standards (Skype, Ekiga, etc) et toutes les bibliothèques standards (OpenCV, Gstreamer) peuvent accéder au capteur. Tous les autres traitements sont effectués avec des bibliothèques de vision standard ou par des programmes OpenCL définis par l'utilisateur sans intervention des concepteurs hardware* », explique Ricardo Rigalda. Peut-être pas pour longtemps… ▮



Le module de contrôle d'E/S de 80 mm x 80 mm est équipé d'un APU T56N de la série G embarquée d'AMD et d'un FPGA.

Cognex fait progresser 2DMax Plus

Pour lire les codes DPM (Direct Part Mark) et matriciels 2D à l'aide de ses Dataman, Cognex proposait déjà la technologie 2DMax Plus. Désormais, l'Américain va encore plus loin avec un nouvel algorithme baptisé Powergrid. Son atout ? Cette technologie permet de lire les codes 2D Datamatrix même lorsqu'ils sont endommagés ou incomplets.

Par construction, un code Datamatrix comprend un motif en « *L* », appelé Finder Patern, qui permet de localiser précisément le code, un autre motif en L en pointillé, qui complète le carré pour définir le nombre de lignes et de colonnes utilisées, et une « *zone de silence* », c'est-à-dire des bandes vierges autour du code, qui l'isolent physiquement. Au milieu, on retrouve le code en lui-même au sein duquel l'information est codée de façon redondante. « *Pour lire l'information, un lecteur traditionnel recherche avant tout ces trois composantes pour localiser le code avant de le déchiffrer. Le Powergrid, lui, est capable de s'en passer* », déclare Benoît Zamboni, ingénieur commercial Identification chez Cognex. L'algorithme de l'Américain va en effet rechercher directement l'information codée dans la grille – d'où le nom grid - de carrés noirs et blancs.

« *Le Powergrid est d'ores et déjà disponible pour les Dataman 300 et 500 actuel dotés du 2Dmax Plus, sous forme d'option* », annonce Benoît Zamboni. La technologie sera également intégrée à tous les nouveaux modèles du fabricant, même d'entrée de gamme. ▮

Le spécialiste de la vision a mis au point une nouvelle technologie qui améliore encore la reconnaissance des codes Datamatrix.