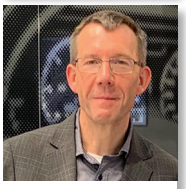
Comment développer de façon homogène sur une puce-système hétérogène

Les puces-systèmes (SoC) hybrides intègrent aujourd'hui de plus en plus de cœurs de processeurs hétérogènes. Cependant, les développeurs les considèrent, de leur point de vue, comme un seul et même composant sur lequel développer, de la manière la plus intégrée possible. Pour y parvenir, il est possible de tirer parti d'un écosystème composé d'un système d'exploitation temps réel et d'un hyperviseur associés de façon homogène, et ce sur les processeurs avec MPU (unité de protection mémoire) et MMU (unité de gestion mémoire).

AUTEUR



Oliver Kühlert, Technical Product Marketing Manager, Sysgo.

Grâce à des processus de fabrication toujours plus petits, il est possible de mettre en œuvre toujours plus d'intelligence et donc une plus grande valeur ajoutée sur une surface silicium donnée. Les fabricants de semi-conducteurs utilisent l'espace ainsi libéré pour intégrer souvent une interface logique native et des cœurs de processeurs hétérogènes pour in fine concevoir des puces-systèmes hybrides (SoC) complètes. L'objectif ici est d'obtenir des appareils électroniques plus réactifs, de plus petite taille et consommant moins d'énergie, dans lesquels on peut intégrer par exemple des systèmes de vision et d'intelligence artificielle. Avec, au-delà de applications dans l'électronique grand public, des applications intégrant des contraintes en matière de sécurité fonctionnelle et susceptibles d'utiliser des SoC hétérogènes. Une situation qui s'accompagne alors généralement d'une nécessaire consolidation des composants logiciels pour réduire les coûts (photo).

Des SoC temps réel pour aller vers la sécurité fonctionnelle

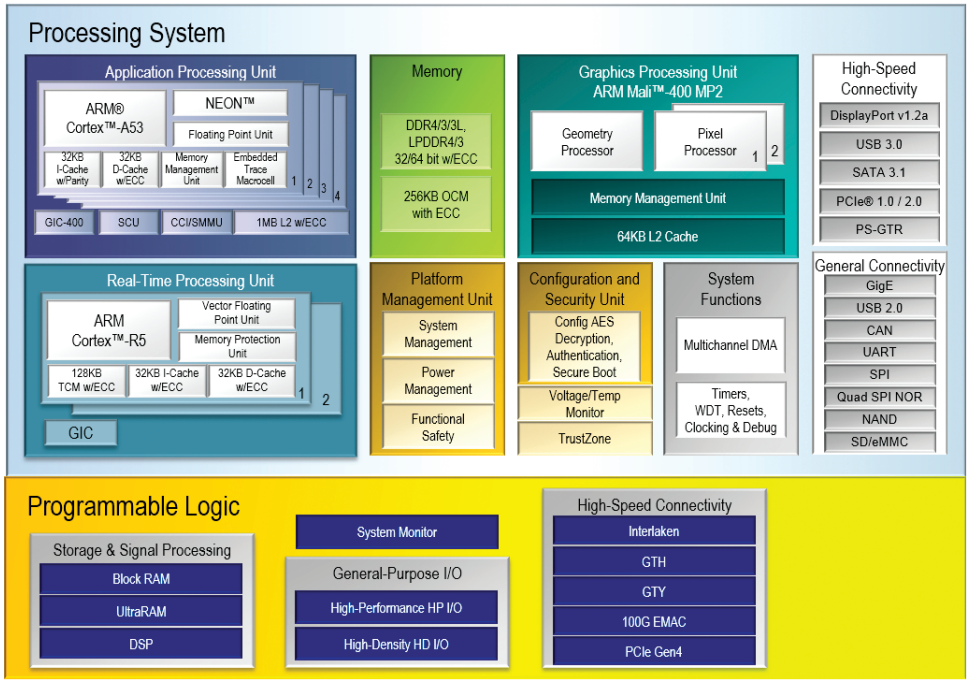
Les marchés des applications qui mettent en œuvre des SoC hétérogènes avec de la sécurité fonctionnelle se trouvent, par exemple, dans les domaines de l'automatisation industrielle, de la robotique collaborative et des technologies médicales certifiées MDR (Medical Device Regulation) qui vont des applications sur des véhicules intralogistiques autonomes jusqu'à des engins de construction et d'agriculture intelligents, en passant par l'électromobilité, le transport ferroviaire et l'avionique. Autant de marchés qui sont des moteurs de croissance pour ces SoC hétérogènes.

La nécessité de disposer de SoC hétérogènes est aussi particulièrement bien illustrée par l'industrie automobile : dans ce secteur, l'intégration

● Dans le secteur de la fabrication industrielle, les véhicules logistiques autonomes et les robots collaboratifs sont des applications reposant souvent sur des SoC intégrés, dotés de technologies de gestion de la mémoire hétérogène. Les machines et les systèmes qui intègrent la sécurité fonctionnelle en plus des IHM et des automates programmables sont d'autres exemples.

1 BLOC-DIAGRAMME DU CIRCUIT MPSOC ZYNQ ULTRASCALE+ DE XILINX

Les MPSoC Xilinx Zynq Ultrascale+ intègrent quatre cœurs Arm Cortex MMU et deux MPU ainsi qu'un puissant FPGA, pour lesquels de nombreux composants (blocs d'IP) prêts à l'emploi sont également disponibles.



d'un grand nombre de calculateurs discrets par véhicule est une pratique courante. On compte par exemple plus de 100 calculateurs répartis dans un véhicule classique de milieu de gamme. Dans les voitures de luxe, on trouve parfois même plus de 300 calculateurs, avec un ensemble de microcontrôleurs 8 ou 32 bits. Toutefois, dans le cadre du développement de l'électromobilité et de la conduite de plus en plus autonome, la conception de systèmes de contrôle-commande est complètement repensée et des systèmes nettement plus intégrés fondés sur des SoC hétérogènes sont de plus en plus souvent mis en œuvre. Avec en ligne de mire la coexistence « pacifique » d'applications avec et sans sécurité fonctionnelle au sein de systèmes dits à criticité mixte.

Les MPSoC Zynq Ultrascale+ de Xilinx (figure 1) sont par exemple une famille de circuits SoC qui peuvent être certifiés pour de telles applications à différents niveaux SIL (Safety Integrity Level), ASIL (Automotive Safety Integrity Level) et DAL (Development Assurance Levels ou Design Assurance Levels), entre autres. Ces SoC intègrent six cœurs Arm, de nombreuses interfaces standard et une matrice de FPGA pour la conception d'applications spécifiques au SoC. Ce qui les rend multifonctionnels et les prédestine même aux solutions de rétrofit des applications existantes. Ils offrent quatre cœurs Arm Cortex-A53 de 64 bits avec une prise en charge complète de l'ECC (Error Correcting Code ou mémoire à code correcteur d'erreurs) et un double cœur Arm Cortex-R5F de 32 bits qui peut être verrouillé ce qui le rend nativement fonctionnellement sûr. Le FPGA dispose de son côté d'entre 81 000 et 504 000 cellules logiques. Un processeur graphique (cœur Mali 400-MP2) est également intégré en option pour le support d'applications graphiques ou pour faire tourner des algorithmes d'inférence de l'intelligence artificielle. En outre, le support des codecs H.264 et H.265 est proposé en option. Les domaines d'application qui en résultent sont variés et comprennent les systèmes sécurisés tels que les contrôleurs du mouvement, les systèmes graphiques et/ou basés sur l'IA, ainsi que les systèmes avec caméra intégrée devant traiter extrêmement rapidement des codecs vidéo pour la connaissance fine d'une situation, voire des algorithmes de réalité augmentée, déterministe en temps réel.

Robots collaboratifs et conduite autonome : le défi de la complexité

Dans un robot collaboratif ou un véhicule intralogistique autonome par exemple, ce SoC peut connecter une caméra au FPGA intégré pour traiter les données d'image. Les données ainsi traitées peuvent ensuite être analysées par un système d'analyse d'images en temps réel installé dans le Cortex-R5F avec une logique d'inférence intégrée pour détecter les obstacles, par exemple. En fonction de la logique de décision intégrée, les résultats peuvent être envoyés à l'Arm Cortex-A53 le plus puissant à travers le canal de communication interne. C'est là que s'effectue le plus haut niveau de logique de commande en temps réel pour le déplacement autonome. En cas de risque de collision de données, le sens de la circulation est modifié en fonction

MMU VS MPU

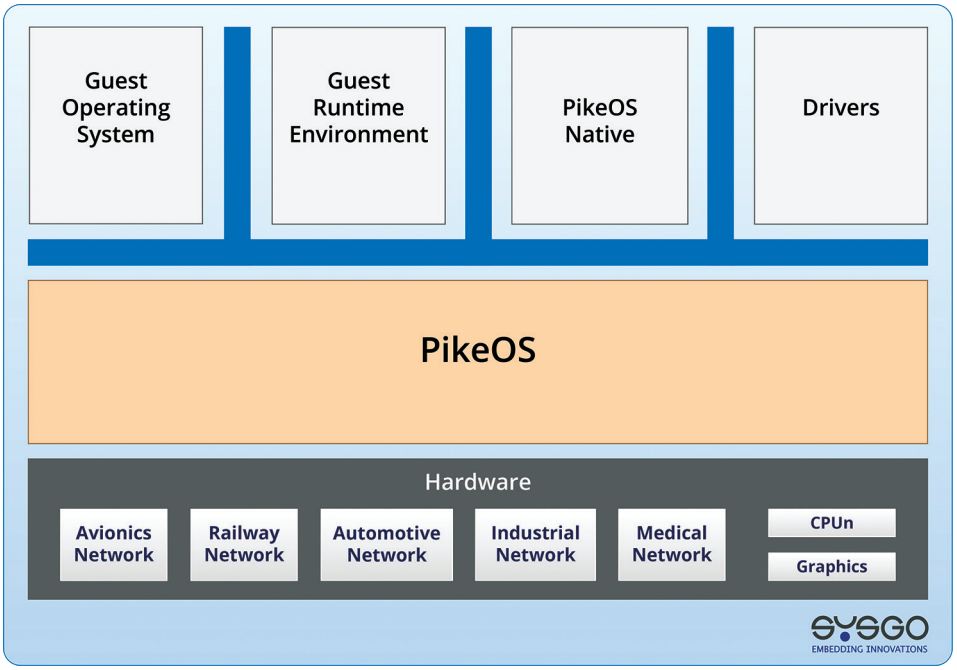
■ La MMU (Management Memory Unit ou unité de gestion de la mémoire) et la MPU (Management Protection Unit ou unité de protection de la mémoire) sont des dispositifs utilisés par un processeur pour la gestion de la mémoire. La MMU est utilisée principalement comme mémoire virtuelle -c'est-à-dire avec un mécanisme de traduction d'une adresse virtuelle en adresse physique- et la protection de la mémoire alors que la MPU est utilisée uniquement pour la protection de la mémoire.

■ Les processeurs qui exécutent des systèmes d'exploitation généralistes comme Linux ou Windows ont généralement une MMU alors que les processeurs ou microcontrôleurs qui exécutent des systèmes d'exploitation temps réel (RTOS), comme ThreadX, ont généralement une MPU. On trouve ainsi de plus en plus d' architectures hétérogènes dans les systèmes embarqués avec un processeur (doté de un ou plusieurs cœurs Arm Cortex-A par exemple) qui exécute un système d'exploitation lourd comme Linux sur lequel des applications sont exécutées et un second (doté de un ou plusieurs cœurs Arm Cortex-R par exemple) sur lequel des piles de protocoles sont exécutées.

■ La mise en œuvre de la MMU au sein d'un processeur est beaucoup plus complexe que celle de la MPU. C'est pourquoi de nombreux systèmes embarqués à fortes contraintes temps réel, et qui n'ont pas besoin de mémoire virtuelle mais d'une protection de la mémoire, ont un MPU beaucoup plus simple à développer qu'un MMU complet.

2 PIKEOS ET HYPERVISEUR POUR MPU

Les nouveaux systèmes PikeOS et Hypervisor for MPU de Sysgo sont conçus pour les processeurs avec une unité de protection de la mémoire dans lesquels une zone de mémoire dédiée doit être allouée à chaque processus. PikeOS for MPU peut également héberger PikeOS pour les processeurs MMU et d'autres systèmes hôtes dans ses machines virtuelles.



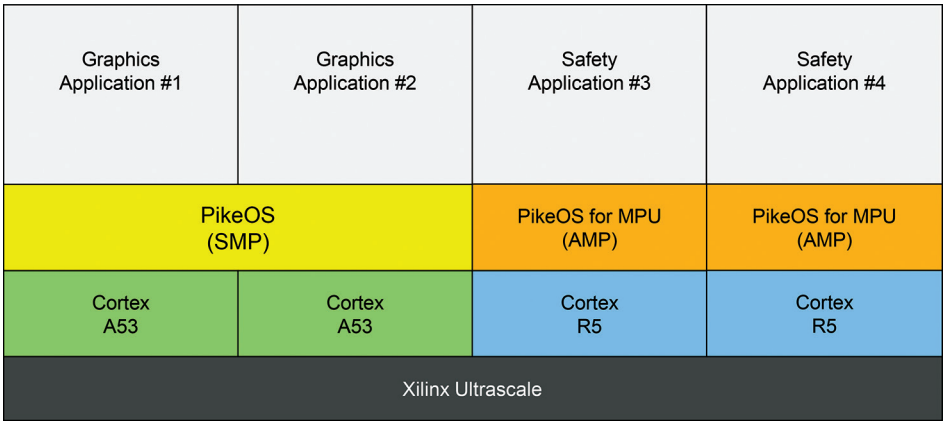
du meilleur itinéraire possible jusqu'à destination. Pour corriger un point unique de défaillance (SPOF, Single Point Of Failure) dans l'évaluation des données d'image, les deux cœurs de l'Arm Cortex-R5F peuvent même fonctionner en mode lockstep. Bien entendu, le cœur Arm Cortex-R5F peut également être utilisé pour des tâches de contrôle en combinaison avec le FPGA. Par exemple, les servomoteurs peuvent être contrôlés directement à travers ces cœurs de processeur. Une autre utilisation intéressante consiste à faire tourner un contrôleur à la fois sur le cœur R5 et le cœur A53 pour créer une redondance de cœurs dans un seul SoC.

Cependant, l'un des plus grands défis de ces conceptions de SoC est le coût élevé du développement et de leur maintenance. Il est donc nécessaire de développer des applications aussi intégrées que possible, car c'est le seul moyen d'obtenir des effets de synergie qui réduisent les coûts NRE (Non Recurring Engineering ou coûts d'ingénierie non récurrents) qui rendent plus efficace le développement de systèmes temps réel aux performances optimales. Cela s'applique également aux inférences de cache dans les implémentations multicœurs, ou bien aux mécanismes d'attribution des tâches d'une application à un cœur ou à un autre selon les besoins, sans avoir à réaliser trop d'efforts de développement.

Sur ce type de circuit hétérogène, on constate que pour les cœurs Arm Cortex-A53 la gestion de la mémoire de travail se fonde sur une unité de gestion de la mémoire (MMU), tandis qu'avec le double cœur Arm Cortex-R5F, la mémoire de travail s'appuie sur une unité de protection de la mémoire (MPU). La différence entre ces différents systèmes de gestion de la mémoire (voir encadré) est qu'une MMU peut être utilisée pour convertir des zones d'adresses virtuelles en zones d'adresses physiques quelconques. La MMU attribue donc une zone d'adresse concrète à un processus. Un contrôleur avec MPU ne dispose pas de cette fonction d'affectation. La MPU offre toujours une protection qui fait qu'un processus ne peut pas écrire à un autre dans la même zone de mémoire, mais dans cette situation chaque processus doit savoir exactement où se lier. Ce qui est conceptuellement plus complexe car chaque processus doit se voir attribuer une zone de mémoire dédiée. Le logiciel d'exploitation temps réel doit alors fournir l'API (Application Programming Interface ou interface de programmation d'application) d'allocation mémoire.

3 ARCHITECTURES PIKEOS SUR LE SOC DE XILINX

Plusieurs PikeOS pour les processeurs avec MMU et MPU peuvent fonctionner en parallèle sur le SoC Xilinx Zynq Ultrascale+ et communiquer de manière transparente entre eux via la communication intercœur.



MMU et MPU : deux mondes à faire cohabiter

Dans ce paysage, l'association d'un système d'exploitation temps réel (RTOS, Real Time Operating System) et d'un hyperviseur temps réel est une voie pour gérer de tels SoC hétérogènes avec des contrôleurs fondés sur des MMU et des MPU. Jusqu'à présent, la plupart des fournisseurs de systèmes d'exploitation ont développé des RTOS plus petits pour les contrôleurs avec MPU avec des API complètement différentes de celles des RTOS pour les contrôleurs avec MMU. Une approche pragmatique qui n'a pas joué un rôle majeur jusqu'ici puisque ces cœurs de pro-

cesseur ont été mis en œuvre de manière discrète, pour la plupart. Cependant, avec des écosystèmes de systèmes d'exploitation homogènes pour le développement de SoC dotés de technologies MMU et MPU, la programmation de ces circuits hétérogènes est désormais à portée des développeurs.

Dans ce cadre, avec le lancement en septembre de cette année du système d'exploitation PikeOS et de l'hyperviseur pour MPU, Sysgo, spécialiste des logiciels embarqués fonctionnellement sûrs et sécurisés, a créé pour la première fois une telle fondation, avec laquelle des SoC hétérogènes bénéficient d'un RTOS et d'un hyperviseur en temps réel homogène, ce qui simplifie considérablement la programmation et l'équilibrage de charges utiles. PikeOS for MPU a été développé à cet effet au niveau du code sur la base du système d'exploitation PikeOS pour les processeurs avec MMU.

Les API permettant de programmer des applications pour les processeurs avec MMU ou MPU sont donc quasiment identiques. Pour l'essentiel, seule l'API de gestion de la mémoire a été adaptée en conséquence, et le passage d'une application d'un processeur avec MMU à un autre processeur avec MPU s'effectue en quelques clics en l'espace de quelques minutes malgré la différence de gestion de la mémoire. Plus important encore est que le code pour les deux variantes de noyau (MMU et MPU) peut être certifié de manière similaire. Et à l'avenir les prochaines certifications de PikeOS pour les solutions sur MPU pourront donc s'appuyer sur les certifications SIL 4, DAL A et ASIL D de PikeOS for MMU.

Vers un écosystème homogène pour MMU et MPU

PikeOS (figure 2) et PikeOS for MPU (figure 3) partagent des fonctions centrales importantes, telles que le noyau de séparation ou les mécanismes de partitionnement temporel et spatial, qui ont pu être maintenus à l'identique sur le plan fonctionnel. En séparant strictement les partitions, le noyau de séparation permet le fonctionnement en parallèle de plusieurs applications – des tâches de contrôle simples mais très critiques, aux programmes utilisateurs complexes comportant de nombreuses fonctions. En outre, le noyau de séparation élimine le risque que des erreurs d'application ne touchent d'autres partitions et applications. L'utilisation des mêmes mécanismes de partitionnement temporel et spatial rapproche également PikeOS for MPU de la spécification ARINC 653 pour laquelle PikeOS for MMU a été initialement développé.

La fonctionnalité ICCOM (Inter-Core Communication) des deux dérivés de PikeOS est une caractéristique particulièrement intéressante pour le développement efficace de solutions globales fondées sur des systèmes hétérogènes : cette fonctionnalité permet en effet aux instances PikeOS fonctionnant sur différents cœurs ARM Cortex-A et R de communiquer entre elles via des canaux de communication qui reposent sur des messages, que les cœurs exécutent des systèmes d'exploitation différents ou identiques (figure 4). ICCOM s'appuie ici sur une couche de transport de données symétrique full duplex qui garantit la livraison des messages.

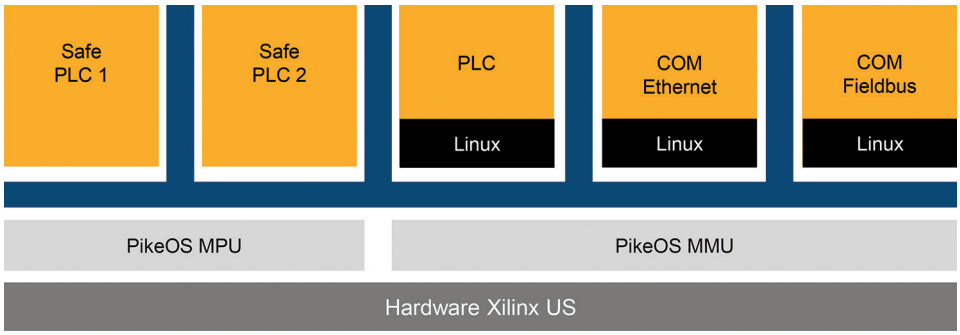
Au-delà, côté programmation et à partir de la version 7.2 de l'environnement de développement intégré Codeo de Sysgo, conçu sur Eclipse (figure 5), les deux systèmes d'exploitation peuvent être utilisés dans un seul environnement de travail. Ce dernier peut gérer l'ensemble de la pile logicielle de SoC hétérogènes et sa communication intercœur au sein d'un seul espace de travail, ce qui simplifie considérablement le processus de développement logiciel pour des systèmes cibles aussi complexes. L'ensemble du cycle de développement est pris en charge, depuis l'émulation précoce du système basée sur la machine virtuelle open source QEMU et la simulation des applications, jusqu'au débogage à distance et aux mécanismes de mises à jour logicielles pour les systèmes déployés sur le terrain.

L'environnement TRACE32 de Lauterbach prend également en charge le débogage de cibles fondées sur des MMU et des MPU. Ce qui signifie également qu'une configuration matérielle TRACE32 est suffisante pour déboguer l'ensemble de la plate-forme MPSoC Xilinx Zynq Ultrascale+ avec une configuration hétérogène du système d'exploitation. Les développeurs ont alors accès à un écosystème homogène pour des SoC hétérogènes dotés d'un hyperviseur de type 1 temps réel intégré dans les deux variantes, MPU et MMU, de sorte que de multiples applications isolées en temps et en mémoire, fonctionnellement sûres, peuvent être hébergées dans des machines virtuelles encapsulées.

En démarrant leurs travaux à travers une interface graphique individuelle pour ces partitions du système d'exploitation, les architectes logiciels peuvent également déboguer simultanément les deux variantes de PikeOS, y compris les événements de démarrage/arrêt synchronisés. Une approche utile lors de la recherche d'erreurs dans la communication entre les différents sous-systèmes. De plus, TRACE32 peut tracer

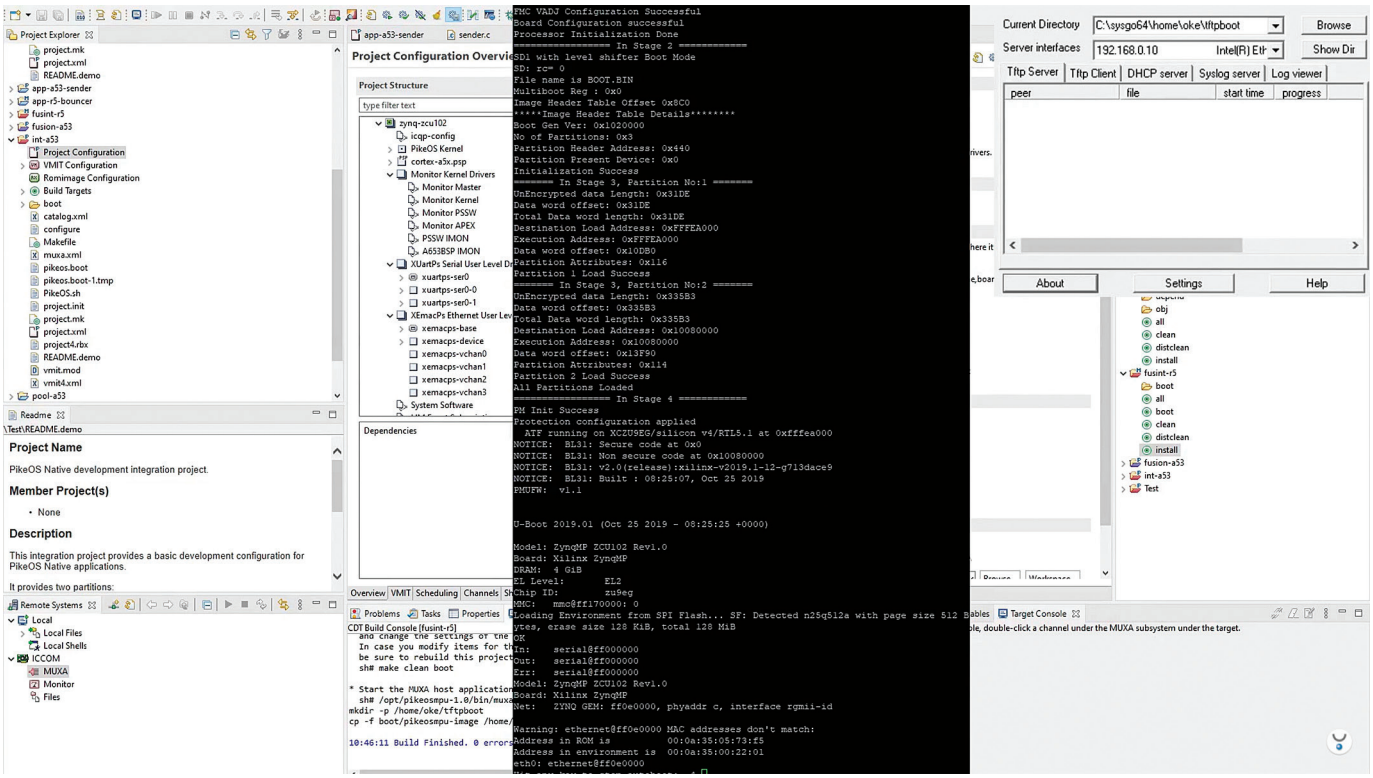
4 ARCHITECTURE PIKEOS + ELINOS

Pour les applications industrielles critiques mixtes qui déploient également un système Linux en plus de PikeOS pour MMU et MPU, il est recommandé d'utiliser le système d'exploitation ELinOS. Dans ce cas, il est possible d'utiliser l'environnement de développement intégré Codeo de Sysgo pour toutes les applications.



5 IDE CODEO

Avec l'environnement de développement intégré Codeo, les images de tous les systèmes d'exploitation Sysgo peuvent être développées de manière intégrée et transférées au système cible en une seule opération.



l'ensemble du système et afficher des graphiques des temps d'exécution des applications et des fonctions. Le timing est synchronisé, ce qui permet d'observer le comportement temporel de PikeOS et de PikeOS for MPU et de mesurer les latences entre les deux systèmes, facilitant ainsi l'équilibrage des performances.

Ces solutions ne sont pas limitées au Xilinx MPSoC Zynq Ultrascale+ mais prennent en charge les cœurs Arm Cortex-A53 et R5 dans n'importe quel circuit, avec la notion de portabilité qui en découle. A moyen terme, il est également prévu de prendre en charge toutes les autres variantes des cœurs Arm Cortex-A, R et M et d'étendre la prise en charge aux processeurs RISC-V, MPC 57xx et TriCore AURIX – en définitive, tous les processeurs discrets basés sur des MPU et les SoC MMU/MPU hétérogènes.

La sécurité fonctionnelle est visée

Des kits de certification pour PikeOS for MMU sont disponibles pour l'avionique (DO-178C), l'automobile (ISO 26262), le ferroviaire (EN 50128 / EN 50657), l'automatisation industrielle (CEI 61508) et le médical (CEI 62304), par exemple. En outre, les domaines d'application visés se situent jusqu'au niveau élevé d'assurance d'évaluation EAL3+ – PikeOS étant le seul noyau de séparation (version 4.2.4) au monde à détenir un certificat Critères Communs actuel et valide au niveau EAL3+ – ou jusqu'au niveau SIL 4 selon la norme de certification de sécurité utilisée. L'écosystème PikeOS constitue également une base adaptée pour la gestion des protocoles OPC UA synchronisés à travers la technologie TSN qui orchestrent les données des véhicules autonomes et/ou de la robotique collaborative connectés à des serveurs dans le cloud (Edge Cloud) grâce à la 5G.

Dans ce contexte, la possibilité d'utiliser la plate-forme de connectivité automobile sécurisée (SACoP) de Sysgo pour les communications de voiture à voiture et de voiture à infrastructure est également adaptée pour la gestion de systèmes mobiles. Elle protège l'infrastructure interne critique du véhicule du monde extérieur à l'aide de pare-feu et de mécanismes de détection des intrusions, et grâce à la mise en œuvre d'une fonctionnalité de démarrage sécurisé, protège également contre les compromissions du système d'exploitation et l'installation de chargeurs de démarrage manipulés. Il prend également en charge les mises à jour OTA (over-the-air) sécurisées ainsi que de nombreuses autres tâches de gestion logicielle, telles que les mises à niveau pour les licences économiques par abonnement basées sur les fonctionnalités. Au fur et à mesure de l'émergence de nouveaux projets clients, tous ces éléments peuvent également être portés sur PikeOS for MPU.

Les installations existantes offrent également un haut niveau de sécurité de conception. Par exemple, le RTOS PikeOS for MPU est déjà utilisé dans des applications spatiales. L'architecture utilisée ici s'appuie sur une puce discrète propriétaire à base d'Arm Cortex-R5 qui est durcie contre les radiations afin d'atténuer les aléas logiques (SEU, Single-Event Upset) induits par les radiations. ■