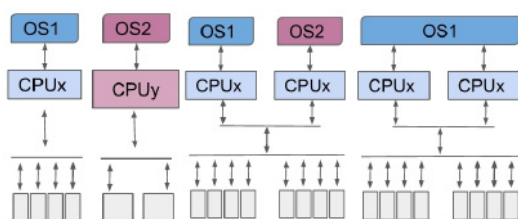


Systemy heterogeniczne

Na rynku systemów komputerowych można spotkać wiele różnorodnych rozwiązań, których celem jest udoskonalenie systemu. Niektóre z nich mają za zadanie zwiększyć wydajność układu, inne zmniejszyć moc pobieraną przez system. Oczywiście nie sposób wymienić wszystkich przykładów z uwagi na fakt, że rozwiązań takich jest wiele i ciągle pojawiają się nowe. Jednym z nich jest umieszczenie kilku różniących się od siebie architektur w jednym układzie scalonym. Niniejszy artykuł przybliży tematykę systemów heterogenicznych i podzielony jest na kilka części. Pierwsza opisuje przypadek systemu, gdzie zastosowano zarówno architekturę ARMv7, jak i Cortex-M3. Kolejna przybliży Linuksowy koncept sterowników typu MISC (ang. Miscellaneous device drivers), który może być użyty do zarządzania rdzeniem Cortex-M3 z poziomu rdzenia ARMv7. Ostatnia część skupia się na wskazówkach dotyczących komunikacji międzyprocesorowej.

PODZIAŁ SYSTEMÓW WIELOPROCESOROWYCH

Biorąc pod lupę systemy heterogeniczne, warto porównać je z innymi systemami wieloprocessorowymi. Jak pokazano na Rysunku 1, wieloprocessorowe systemy można podzielić na trzy różne grupy: SMP (ang. *Symmetric Multiprocessing*), AMP (ang. *Asymmetric Multiprocessing*) oraz systemy heterogeniczne (ang. *Heterogenous systems*). Poniżej zamieszczono krótki opis poszczególnych grup.



Rysunek 1. Podział systemów wieloprocessorowych (od lewej: SMP, AMP, Systemy heterogeniczne)

W systemach SMP znajdują się co najmniej dwa rdzenie tego samego typu. Do ich obsługi wykorzystywany jest jeden OS, wspierający wielordzeniowość. Jego zadaniem jest wykorzystanie mocy obliczeniowej wszystkich jednostek wykonawczych poprzez odpowiednie przydzielanie zadań do poszczególnych rdzeni, jak również zapewnienie synchronizacji między nimi.

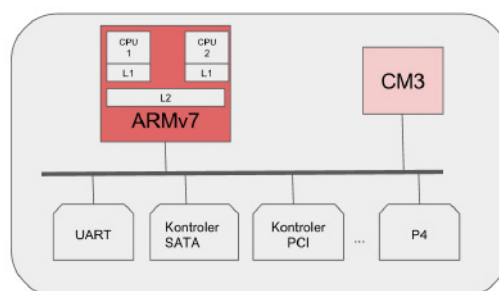
Systemy AMP, podobnie jak systemy SMP, zbudowane są z co najmniej dwóch rdzeni tego samego typu. Jednak w tym przypadku na każdym z rdzeni uruchomiony jest albo różny system operacyjny, albo inna instancja tego samego systemu operacyjnego. W rozwiązaniach takich rdzenie pracują niezależnie od siebie, a poszczególne systemy operacyjne zwykle nie synchronizują się ze sobą poza przypadkami, w których wymieniają wyniki swoich obliczeń.

Z kolei systemy heterogeniczne rozszerzają niejako definicję systemów AMP, pozwalając na współdziałanie rdzeni różnego typu. Przykłady takich architektur często można znaleźć w tzw. układach scalonych typu SoC (ang. *System on Chip*), które wbudowują kompletny system w jednym chipie.

PRZYKŁADOWY SYSTEM HETEROGENICZNY

Typowy układ typu SoC składa się z procesora jedno- lub wielordzeniowego oraz różnego rodzaju urządzeń peryferyjnych i kontrolerów. Można w nim znaleźć wiele różnorodnych pod-modułów, począwszy od interfejsu portu szeregowego UART, a skończywszy na bardziej zaawansowanych układach, jakimi są kontrolery sieciowe, PCI czy też SATA.

Jak już wspomniano wcześniej, niektóre z dostępnych na rynku układów typu SoC posiadają co najmniej dwa procesory różnego typu, a co za tym idzie – są przykładami systemu heterogenicznego. Na Rysunku 2 przedstawiono przykładowy schemat blokowy, który w jednym układzie scalonym mieści zarówno procesor ARMv7, jak i Cortex-M3.



Rysunek 2. Schemat blokowy przykładowego układu typu SoC

Umieszczenie Cortexa-M3 w jednym układzie scalonym obok procesora ARMv7, urządzeń peryferyjnych oraz kontrolerów nie jest kosztowne z punktu widzenia zajmowanego krzemu. CM3 stanowi niewielką część SoCa, a jednocześnie rozszerza jego możliwości. Dodatkowy procesor może być wpięty w tę samą szynę danych, w którą wpięte są urządzenia peryferyjne oraz główny procesor. Podłączenie takie umożliwia Cortexowi-M3 niezależne komunikowanie się z pozostałymi blokami funkcjonalnymi, bez konieczności ingerencji ARMv7. Oczywiście bezpośrednia komunikacja z głównym procesorem także jest możliwa.