

Zastosowania procesorów sygnałowych

***ALTERNATYWNE PROCESORY
DO PRZETWARZANIA
SYGNAŁÓW***

Opracowanie: Grzegorz Szwoch

Politechnika Gdańska, Katedra Systemów Multimedialnych

Wprowadzenie

- W dawnych czasach podział układów mikroprocesorowych był wyraźny.
 - Mikrokontrolery – uproszczone, do sterowania urządzeniami.
 - Mikroprocesory – ogólnego przeznaczenia, do wykonywania programów.
 - Procesory sygnałowe – wyspecjalizowane w przetwarzaniu sygnałów (inna architektura, szerszy zestaw instrukcji procesora, itp.).
- Z biegiem czasu te różnice zaczęły się zacierać.
- Cechy specyficzne dla procesorów sygnałowych zaczęły przenikać do procesorów ogólnego przeznaczenia (CPU).
- Przykład: rozszerzenia SSE dla procesorów Intel (np. instrukcja MAC, wektoryzacja).
- Popularyzacja procesorów w architekturze ARM sprawiła, że procesory te stały się ciekawą alternatywą dla klasycznych procesorów sygnałowych, których rozwój jest znacznie wolniejszy.

Procesory ARM

- Procesory w architekturze ARM są obecnie szeroko stosowane: w smartfonach, tabletach, mikrokomputerach (np. *Raspberry Pi*), systemach wbudowanych.
- Dobra moc obliczeniowa przy niższym zużyciu energii niż w procesorach Intel.
- Rodzaje (profile) procesorów ARM typu *Cortex*:
 - Cortex-A (*Application*) – aplikacje i systemy operacyjne (np. w smartfonach),
 - Cortex-M (*Microcontroller*) – sterowanie układami elektronicznymi,
 - Cortex-R (*Real-time*) – do systemów czasu rzeczywistego.
- Wybrane, obecnie używane architektury:
 - ARMv7, 32-bitowa, obecnie głównie profil M (Cortex-M4, Cortex-M7).
 - ARMv8, 32- i 64-bitowa, najczęściej stosowana w obecnych profilach A, np. *Raspberry Pi 5*: Cortex-A76).

Procesory ARM

Najważniejsze cechy procesorów ARM:

- dobra wydajność obliczeniowa (wyższa w profilu A niż w M),
- niskie zużycie energii, niski poziom emisji ciepła,
- procesory mogą być wielordzeniowe (np. *Raspberry Pi*: 4 rdzenie),
- większość ma jednostkę zmiennoprzecinkową FPU (poza niektórymi M),
- posiadają instrukcje pomocne w przetwarzaniu sygnałów, np. MLA („MAC”),
- umożliwiają wektoryzację obliczeń (zrównoleglanie SIMD),
- mogą posiadać jednostki do sprzętowego przyspieszenia obliczeń na liczbach zmiennoprzecinkowych (VFP, rozszerzenie *Neon*),
- możliwość uruchamiania programów w trybie *bare metal*,
- niski koszt w porównaniu ze zmiennoprzecinkowymi procesorami sygnałowymi oraz z procesorami Intel.

Przykład procesora ARM

Układ VAR-SOM-6UL firmy *Variscite*

- procesor *Freescale i.MX6* w architekturze ARM Cortex-A7,
- jeden rdzeń procesora,
- maksymalna częstotliwość procesora 792 MHz,
- sprzętowe przyspieszenie obliczeń dla liczb zmiennoprzecinkowych 32-bitowych (typ *float*) i całkowitych za pomocą NEON,
- płytkę zawiera 256 MB pamięci DDR3 oraz interfejsy: USB, wejście/wyjście dźwiękowe, Ethernet, I²C, SPI, CAN, UART, JTAG.

Układ został wykorzystany w projekcie badawczym jako element Interkomu – oprogramowanie wykrywa zdarzenia akustyczne i dokonuje ich parametryzacji dla algorytmów uczenia maszynowego (rozpoznawanie zdarzeń).



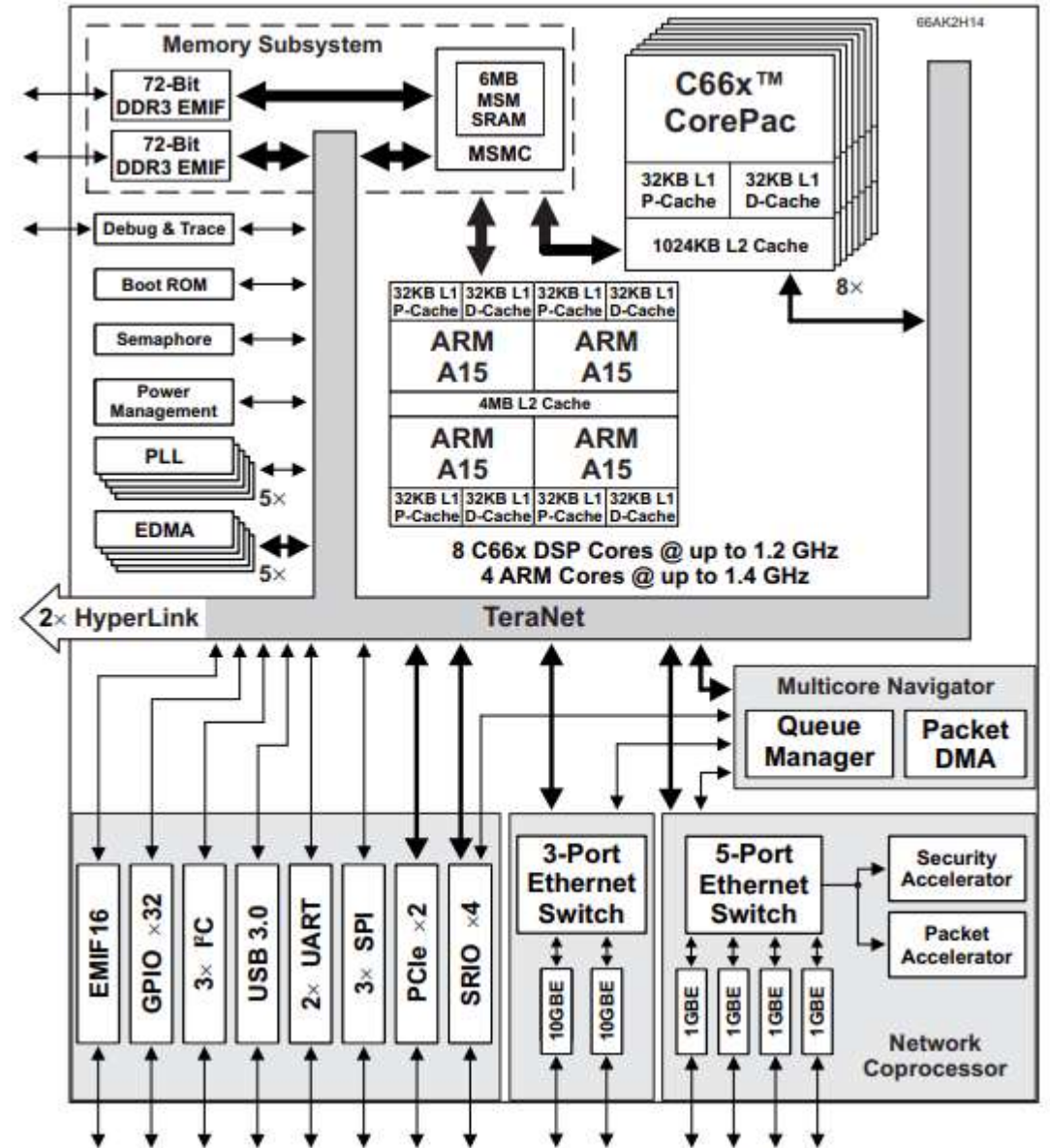
Hybrydowe procesory sygnałowe

- Hybrydowe procesory sygnałowe łączą w jednym układzie:
 - rdzenie (jeden lub kilka) procesora w architekturze ARM,
 - jeden lub więcej rdzeni procesora sygnałowego, zwykle zmiennoprzecinkowego.
- Na głównym rdzeniu uruchamiany jest system operacyjny, np. Linux z jądrem RT.
- Program na głównym rdzeniu zarządza obliczeniami oraz wykonuje operacje wejścia/wyjścia.
- Rdzenie procesora sygnałowego wykonują obliczenia.
- Jeżeli mamy wiele rdzeni, możemy zrównoleglić obliczenia, np. podzielić klatkę obrazu na kilka części, każdy rdzeń przetwarza swoją część.
- Wymiana danych między ARM a DSP za pomocą systemu kolejek.
- Duży narzut obliczeń – zarządzanie wątkami, synchronizacja.
- Duża elastyczność, ale skomplikowany sposób tworzenia programu.

Procesory hybrydowe - przykład

66AK2H12 *Keystone* (Texas Instruments):

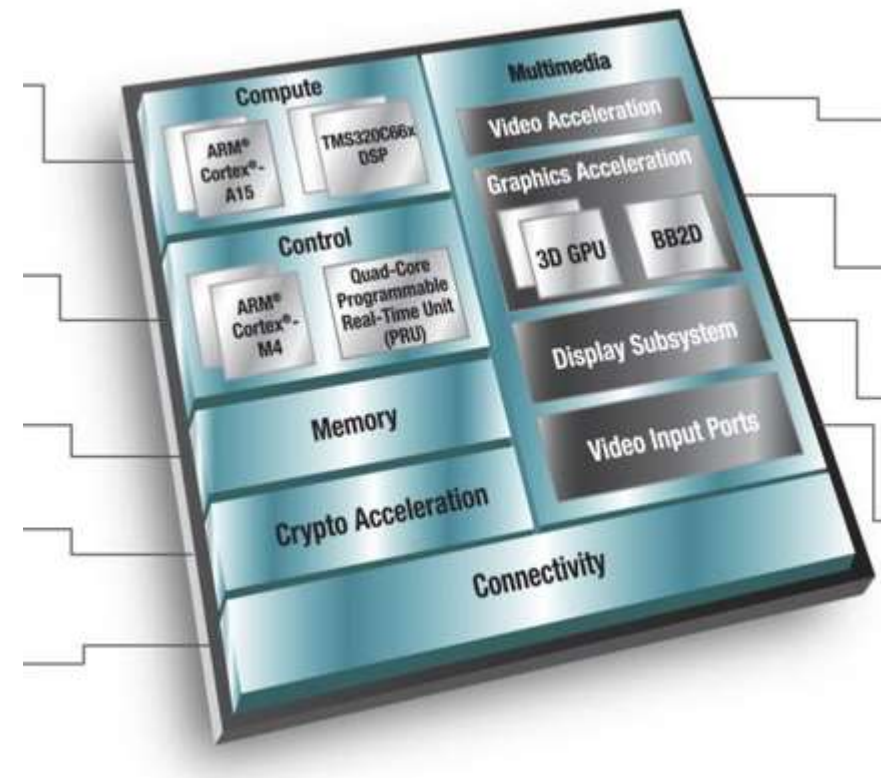
- 4 rdzenie w architekturze ARM Cortex-A15,
- 8 rdzeni procesora sygnałowego C66x.



Procesory obrazu

Procesor obrazu – *Digital Image Processor* lub *Image Signal Processor (ISP)*

- Procesor wyspecjalizowany w przetwarzaniu cyfrowych obrazów 2D.
- Cyfrowy obraz: specyficzny typ dwuwymiarowego sygnału.
- Wydajny dostęp do pikseli obrazu w pamięci - według rzędów i kolumn.
- Szybkie obliczanie splotu 2D (filtracja obrazu).
- Typowe zastosowania: kamery systemów monitoringu, kamery w pojazdach autonomicznych, dronach, robotach, itp. – analiza zawartości obrazu.
- Przykłady: procesory Texas Instruments: OMAP, Sitara, DaVinci.



Procesory obrazu

Procesory obrazu są wykorzystywane do kontekstowej analizy zawartości obrazu (*video content analysis*):

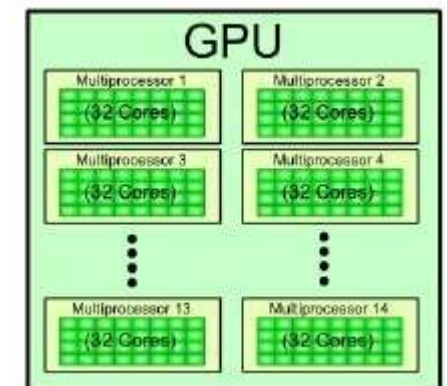
- wykrywanie i śledzenie ruchomych obiektów w kamerze,
- rozpoznawanie obiektów (osoby, pojazdy),
- wykrywanie niebezpiecznych zdarzeń (wandalizm, kradzież),
- sklepy bezobsługowe,
- systemy samochodowe (np. rozpoznawanie znaków drogowych),
- kierowanie autonomicznymi pojazdami,
- sterowanie UAV (dronami), rozpoznawanie obiektów,
- i wiele innych.



Procesory GPU

GPU (*Graphic Processing Unit*)

- Układ zoptymalizowany do operacji związanych z tworzeniem grafiki 3D.
- Procesor **równoległy** – wiele (nawet kilkaset) jednostek obliczeniowych („rdzeni”), wykonujących jednocześnie te same operacje na różnych danych (SIMD).
- Uruchamianie programów: *vertex shader* (operacje na wierzchołkach modeli 3D obiektów), *pixel shader* (operacje na pikselach obrazu).
- Równoległa architektura GPU jest wykorzystywana do **zrównoleglania obliczeń**.
- Przykłady zastosowania: pojazdy autonomiczne (analiza obrazu z kamer), uruchamianie sieci neuronowych, „kopanie kryptowalut”.
- Duże zużycie energii, wysoki koszt układu.
- Duży narzut obliczeniowy przetwarzania.



Procesory GPU - Jetson Nano

Przykład układu GPU: *Nvidia Jetson Nano*

- CPU: ARM Cortex-A57 (4 rdzenie),
- GPU: *Nvidia Maxwell*, 128 rdzeni,
- pamięć 4 GB,
- sprzętowe kodowanie i dekodowanie wideo,
- interfejsy: kamera, Ethernet, USB, HDMI, GPIO, I2C, I2S, SPI, UART,
- system operacyjny: np. Linux Ubuntu,
- maksymalny pobór mocy: ok. 10 W.



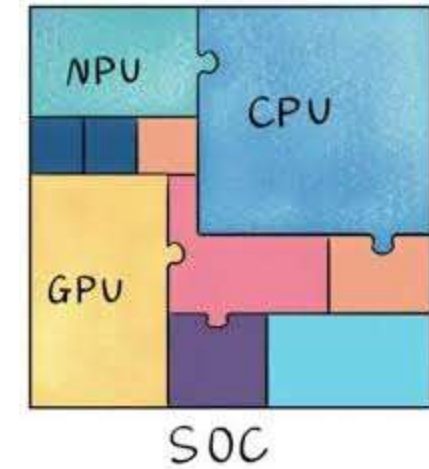
Procesory NPU

- **NPU – *Neural Processing Unit***, inaczej *AI Accelerator*.
- Układ przeznaczony do wykonywania operacji związanych z uczeniem maszynowym, głównie do uruchamiania (inferencji) sieci neuronowych.
- Może być samodzielnym układem lub jednostką składową procesora.
- Architektura wielowątkowa.
- Wydajne wykonywanie operacji takich jak mnożenie macierzy, funkcje aktywacji neuronów, splot 2D, itp.,
- Mniejsze zużycie energii i niższy koszt niż procesory GPU.
- Wydajniejsze przetwarzanie związane z sieciami neuronowymi niż GPU i CPU.
- Układy GPNPU – hybryda GPU i NPU.
- Trening sieci: GPU. Inferencja: CPU + NPU.

Układy SoC

SoC – System on a Chip

- Układ elektroniczny integrujący wiele komponentów w jednej „kostce”:
 - główny mikroprocesor (CPU),
 - procesor graficzny GPU (często),
 - pamięć (różnego typu),
 - procesor sygnałowy DSP (niektóre),
 - dodatkowe koprocesory (np. sprzętowe kodowanie i dekodowanie obrazu, NPU),
 - interfejsy wejściowe/wyjściowe.
- Praktycznie pełny mikrokomputer w jednym układzie.
- Układy typu SoC to obecny trend na rynku technologicznym, zwłaszcza w urządzeniach mobilnych i w systemach wbudowanych.



Przykład układu SoC

Układ *VAR-SOM-MX8M-PLUS* firmy *Variscite*:

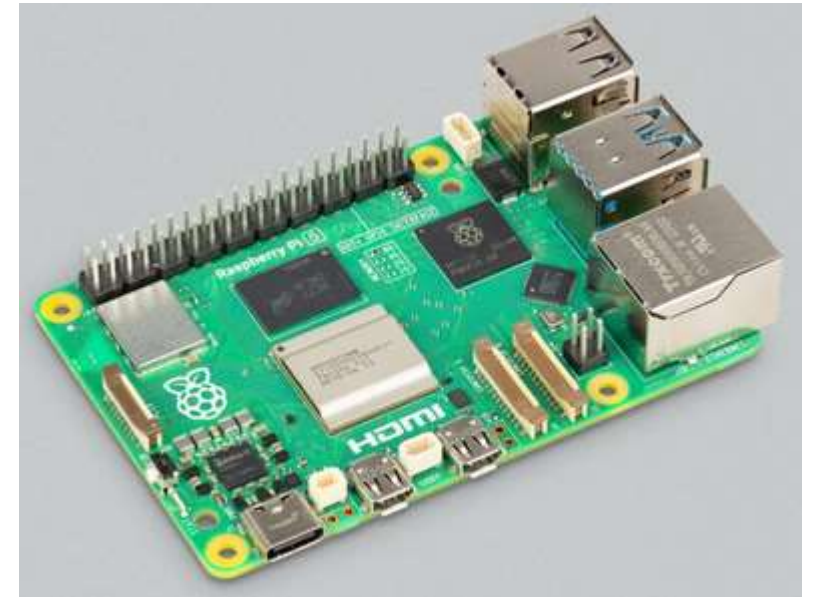
- procesor ARM Cortex-A53 (ARMv8-A),
- 4 rdzenie procesora, maksymalna częstotliwość 1,8 GHz,
- koprocesor ARM Cortex-M7 (800 MHz) – może być wykorzystywany jako mikrokontroler, bez angażowania głównego procesora (oszczędzanie energii),
- koprocesor NPU (*Neural Processing Unit*) o wydajności 2,3 TOPS, do zastosowań związanych z uczeniem maszynowym,
- koprocesor VPU (*Video Processing Unit*),
- akceleratory grafiki 3D i 2D,
- układ „Hifi4 Audio DSP” (do 800 Hz).



Przykład układu SoC

Mikrokomputer *Raspberry Pi 5*: układ SoC *Broadcom BCM2712*

- procesor CPU: ARM Cortex-A76 @ 2,4 GHz, 4 rdzenie,
- procesor GPU *Video Core VII*,
- pamięć RAM DDR4 4 GB lub 8 GB,
- Wi-Fi 802.11ac, Bluetooth 5.0, Ethernet,
- interfejsy: GPIO, PCIe, MIPI (kamera), USB, HDMI,
- maksymalny pobór mocy: ok. 12 W (w praktyce: ok. 6,4 W),
- system operacyjny Linux Raspbian.



Układy FPGA

FPGA (*Field Programmable Gate Array*)

- Układ, którego architektura **sprzętowa** może być konfigurowana przez programistę.
- Macierz programowalnych bloków logicznych z pamięcią – połączenia między nimi mogą być modyfikowane.
- Architektura może zostać zoptymalizowana pod kątem danego algorytmu.
- Architektura procesora jest programem, z wbudowaną pamięcią.
- FPGA zastępują procesory sygnałowe w rozwiązaniach, w których wydajność tych procesorów jest niewystarczająca (obliczenia równoległe, przetwarzanie z bardzo dużą częstotliwością).
- Duża elastyczność – można dostosować architekturę do programu.
- Bardzo wysoki koszt.
- Skomplikowane programowanie, specjalne języki (nie C), np. VHDL.

Układy ASIC

ASIC (*Application-Specific Integrated Circuit*)

- Specjalizowany układ scalony, którego architektura jest fabrycznie zoptymalizowana pod kątem konkretnego zastosowania.
- Nie ma możliwości modyfikacji architektury, takiej jak w FPGA.
- Duża (optymalna) wydajność w jednej konkretnej aplikacji.
- Oprogramowanie (*firmware*) zawarte w pamięci układu.
- Są zwykle szybsze niż FPGA i procesory sygnałowe, zużywają mniej mocy.
- Bardzo mała elastyczność – tylko jedno wybrane zastosowanie.
- Duże koszty projektowania i produkcji.
- Zastosowania: w masowo produkowanych urządzeniach, np. cyfrowe aparaty fotograficzne i kamery, synteзаторы muzyczne.

Układy ASIC

Przykład układu ASIC: procesor stosowany w cyfrowym aparacie fotograficznym, np. *Canon DIGIC*, najważniejsze wykonywane funkcje:

- demozaiikowanie danych z matrycy światłoczułej,
- ustawianie parametrów ekspozycji – czułość ISO, przysłona, czas naświetlania,
- automatyczne ustawianie ostrości (*autofocus*),
- redukcja szumu w obrazie,
- korekcja balansu bieli w obrazie,
- stabilizacja obrazu (optyczna i cyfrowa),
- automatyczny wybór programu,
- wyostrzanie obrazu, efekty (filtry),
- tryb HDR, *exposure bracketing* (kilka zdjęć z różnym EV),
- kompresja obrazu (np. JPEG) do zapisu na nośniku.



Który układ wybrać?

Mamy kilka możliwości procesorów do cyfrowego przetwarzania sygnałów.

Który z nich wybierzemy?

- Tradycyjne **procesory sygnałowe** (DSP) są stosowane coraz rzadziej.
 - Przydają się gdy potrzebujemy prostego przetwarzania, typu filtracja (usuwanie zakłóceń), FFT, splot, itp. – te procedury są bardzo wydajne.
 - Ograniczona wydajność, niskie zużycie energii.
 - Dość kłopotliwe programowanie, zwłaszcza na procesorach stałoprzecinkowych.
 - Są mniej elastyczne niż układy oparte na procesorach ARM, które charakteryzują się podobną ceną, wydajnością i zużyciem energii.

Który układ wybrać?

- Układy SoC oparte na procesorach w architekturze ARM.
 - Najbardziej elastyczne pod względem zastosowań.
 - Dobra wydajność, dostępne są np. procesory czterordzeniowe.
 - Przystępna cena, niskie zużycie energii.
 - Wygodne programowanie, np. używając systemu Linux.
 - Możliwość uruchamiania wielu procesów z wieloma wątkami.
 - Możemy wybrać sobie układ SoC zawierający potrzebne nam koprocesory, np. GPU, NPU, mikrokontroler, itp., nawet procesor sygnałowy.
 - Lepsza wydajność i większe możliwości niż procesory sygnałowe w zbliżonej klasie cenowej.
 - Układy SoC oparte na ARM są obecnie najczęściej używanymi platformami do przetwarzania sygnałów w systemach wbudowanych.

Który układ wybrać?

- Układy zawierające procesor graficzny GPU.
 - Przydatne do przetwarzania danych w trybie SIMD, gdy możemy zrównoleglić obliczenia, a przez to przyspieszyć je.
 - Stosowane głównie do analizy obrazu z kamer wysokiej rozdzielczości oraz do algorytmów uczenia maszynowego.
 - Stosunkowo wysoka cena i zużycie energii.
- Układy zawierające procesor NPU.
 - Do uruchamiania algorytmów opartych na sieciach neuronowych (inferencja).
 - Większa wydajność i niższe zużycie energii w porównaniu z GPU.
 - Jeżeli potrzebne jest douczanie sieci neuronowych, potrzebujemy układu GPNPU lub osobnego procesora GPU.

Który układ wybrać?

- Układy **FPGA**.
 - Stosowane w aplikacjach w których zależy nam zarówno na wysokiej wydajności, jak i na możliwości dostosowywania architektury do programu.
 - Wysoka elastyczność i wydajność.
 - Są bardzo drogie.
 - Programowanie jest trudne.
- Układy **ASIC**.
 - Stosowane głównie w firmach wytwarzających masowe produkty na rynek.
 - Lepsza wydajność i zużycie energii niż przy użyciu normalnych procesorów ARM.
 - Bardzo mała elastyczność, gdy układ jest już wykonany.
 - Wysoki koszt projektowania i produkcji (przerzucany na klienta).