

Zastosowania Procesorów Sygnałowych

Adam Korzeniewski - adamkorz@sound.eti.pg.gda.pl, p. 732
dr inż. Grzegorz Szwoch - greg@sound.eti.pg.gda.pl, p. 732
dr inż. Piotr Ody - piotrod@sound.eti.pg.gda.pl, p. 730

Wprowadzenie

Plan wykładu

- Plan przedmiotu ZPS
- Cele nauczania
- Przetwarzanie cyfrowe czy nie?
- Przetwarzanie w czasie rzeczywistym
- Dlaczego potrzebujemy procesorów DSP
- Typowe algorytmy
- Procesory zmiennie- a stałoprzecinkowe

Plan przedmiotu ZPS

- Wprowadzenie - *AK*
- Procesor sygnałowy czy komputer? - *AK*
- Architektura typowego PS i systemy wieloprocessorowe - *AK*
- Bufory kołowe i szybki splot - *AK*
- Liczby stałe- i zmiennoprzecinkowe - *AK*
- Rynek procesorów sygnałowych i zastosowania w fotografii cyfrowej – *AK*
- Procesory sygnałowe w aparatach słuchowych (ale nie tylko) - *PO*

Plan przedmiotu ZPS

- Wprowadzenie do programowania na procesorze sygnałowym - *GSz*
- Filtry FIR i biblioteka DSPLIB - *GSz*
- Filtry IIR - *GSz*
- Przekształcenie Fouriera i splot - *GSz*
- Generowanie sygnałów na procesorze sygnałowym - *GSz*
- Zaawansowane algorytmy DSP - *GSz*
- **Kolokwium**

Dlaczego warto przetwarzać cyfrowo?

- Techniki cyfrowego przetwarzania sygnałów (CPS) są teraz tak silne, że czasem jest bardzo trudno, a nawet niemożliwe jest, aby uzyskać podobne charakterystyki, jak w analogowym przetwarzaniu sygnałów.

Przykłady:

- Filtry cyfrowe FIR (o skończonej odpowiedzi impulsowej) z liniową charakterystyką fazową.
- Filtry adaptacyjne

Cele nauczania

- Dlaczego warto przetwarzać sygnały cyfrowo?
- Definicja przetwarzania w czasie rzeczywistym.
- Dlaczego używamy procesorów DSP (od ang. Digital Signal Processing)?
- Jakie są typowe algorytmy DSP?
- Parametry, jakie warto rozważyć, gdy wybieramy procesor DSP.
- Programowalne DSP w przeciwieństwie do ASIC.

Dlaczego warto przetwarzać cyfrowo?

Z procesorem DSP jest łatwo:

- Zmieniać aplikacje.
- Korygować aplikacje.
- Aktualizować aplikacje

Dodatkowo DSP redukuje:

- Podatność na wpływ szumu.
- Obliczanie układu scalonego (*od ang. chip count*)
- Czas rozbudowy i rozwoju projektu.
- Koszty.
- Zużycie mocy.

Dlaczego NIE cyfrowo?

- Nie można przetwarzać cyfrowo sygnałów wysokiej częstotliwości z dwóch powodów:**
- Konwertery analogowo cyfrowe (ADC) nie mogą pracować dostatecznie szybko (próg jednak się przesuwają)
 - Aplikacja może okazać się zbyt skomplikowana by można było ją zrealizować w czasie rzeczywistym

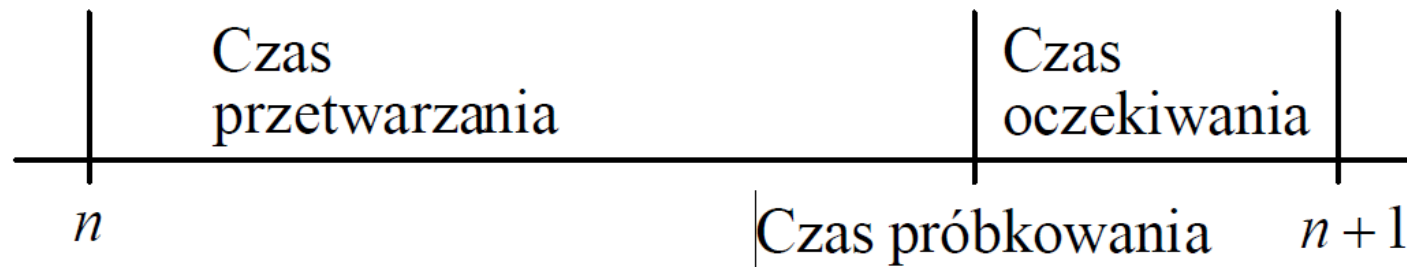
Przetwarzanie w czasie rzeczywistym

- Procesory DSP muszą wykonywać zadania czasie rzeczywistym. Jak definiujemy czas rzeczywisty?
- Definicja czasu rzeczywistego zależy od zastosowania
- Przykład: Filtr FIR o 100 współczynnikach (odczepach) wykonuje filtrację w czasie rzeczywistym, gdy procesor DSP jest w stanie wykonać i zakończyć następującą operację pomiędzy dwoma próbkami:
$$y[n] = \sum_{k=0}^{99} a[k]x[n-k]$$

Przetwarzanie w czasie rzeczywistym

Możemy powiedzieć, że mamy do czynienia z przetwarzaniem w czasie rzeczywistym jeżeli:

- Czas oczekiwania (ang. waiting time) ≥ 0



Dlaczego potrzebujemy DSP?

Użyj procesora DSP gdy potrzebne są:

- Oszczędność kosztów
- Mniejszy rozmiar
- Niski pobór mocy
- Przetwarzanie wielu sygnałów o wysokich częstotliwościach w czasie rzeczywistym

**Użyj procesora GPP (ogólnego przeznaczenia),
gdy potrzebne są:**

- Duża pojemność pamięci
- Zaawansowane systemy operacyjne

Jakie są typowe algorytmy DSP

Suma iloczynów (ang. *The Sum of Products (SOP)*) to kluczowy element w większości algorytmów DSP:

$$\text{Filtr FIR} \quad y[n] = \sum_{k=0}^M a_k x[n-k]$$

$$\text{Filtr IIR} \quad y[n] = \sum_{k=0}^M a_k x[n-k] + \sum_{k=1}^N b_k y[n-k]$$

$$\text{Splot dyskretny} \quad y[n] = \sum_{k=0}^N x[k] h[n-k]$$

$$\text{DFT} \quad X[k] = \sum_{n=0}^{N-1} x[n] \exp[-j(2\pi / N)nk]$$

$$\text{DCT} \quad F[u] = \sum_{x=0}^{N-1} c[u] f[k] \cos\left(\frac{\pi}{2N} u[2x+1]\right)$$

Procesory zmiennoprzecinkowe a stałoprzecinkowe

Aplikacje, które wymagają:

- Wysokiej dokładności (precyzji)
- Szerokiego zakresu dynamicznego.
- Wysokiego stosunku mocy sygnału do szumu
- Łatwości w użyciu

Potrzebują procesora zmiennoprzecinkowego

Wady procesora zmiennoprzecinkowego:

- Większe zużycie mocy
- Mogą być droższe
- Mogą być wolniejsze niż ich odpowiedniki stałoprzecinkowe i o większych rozmiarach.

Procesory zmiennoprzecinkowe a stałoprzecinkowe

- To aplikacja dyktuje, które urządzenie lub platformę należy użyć aby osiągnąć optymalną wydajność przy niskim koszcie.
- Dla celów edukacyjnych użyj procesora zmiennoprzecinkowego (C6711) ponieważ on może realizować zarówno operacje zmiennoprzecinkowe jak i stałoprzecinkowe

Dziękuję za uwagę
