

PG WETiI Katedra Systemów Automatyki

***Procesory sygnałowe i
logika programowalna***

Dr inż. Krzysztof Cisowski

Tel: 583471274, email: www.eti.pg.gda.pl

Kierunek studiów Automatyka i Robotyka

Zakres i treść przedmiotu (1)

1. Definicja i cechy charakterystyczne odróżniające procesory sygnałowe od uniwersalnych mikroprocesorów)
2. Przykłady zastosowań *procesorów sygnałowych* (**DSPs** ang. *Digital Signal Processors*)
3. Omówienie cech charakterystycznych **DSPs** w porównaniu z *uniwersalnymi mikroprocesorami* (**GPSs** ang. *General Purpose Processors*)
4. Klasyfikacja **DSPs**
5. Historia rozwoju **DSPs**
6. Testy wydajności wybranych **DSPs**
7. Producenci procesorów sygnałowych
8. Budowa **DSPs** - przykłady
9. Architektura i instrukcje procesora **TMS320C6713**

Zakres i treść przedmiotu (2)

10. Funkcje modułów ***DSP/BIOS***
11. System rozwojowy ***DSP TMS320C6713***
12. ***Code Composer Studio 3.3 i 4.0 (CCS)***
13. Algorytmy *cyfrowego przetwarzania sygnałów* (***DSP*** ang. Digital Signal Processing)
 - przykłady programów dla procesora ***TMS320C6713***
14. Współpraca ***TMS320C6713*** z innymi platformami: ***Matlab, MFC Visual C++, LabVIEW***
15. Protokół transmisji ***Real Time Data Exchange (RTDX)***
16. Przykłady zastosowań – bardziej szczegółowo niż w pkt. 2
17. Perspektywy rozwoju ***DSPs***

1. Definicja i cechy charakterystyczne odróżniające procesory sygnałowe od uniwersalnych mikroprocesorów

Procesory sygnałowe (DSPs ang. Digital Signal Processors) to układy elektroniczne należące do klasy specjalizowanych procesorów służących do cyfrowej obróbki sygnałów.

Cechy odróżniające DSPs od uniwersalnych mikroprocesorów (GPPs ang. General Purpose Processors):

- rozdzielenie pamięci programu i danych (architektura harwardzka) z możliwością równoczesnego odczytu instrukcji oraz danych,
- sprzętowe dostosowanie do wykonywania operacji najczęściej występujących przy przetwarzaniu sygnałów tj. filtracji **FIR** i **IIR**, transformacji **Fouriera**, obliczaniu korelacji wzajemnej,
- potokowe przetwarzaniem instrukcji,
- specjalne mechanizmy do realizacji operacji wejścia i wyjścia w czasie rzeczywistym,
- niższe zużycie energii oraz niższy koszt zakupu w porównaniu z procesorami ogólnego przeznaczenia.

2. Przykłady zastosowań procesorów sygnałowych - główne zastosowania DSPs

- Cyfrowa telefonia komórkowa
- Telefonia **VOIP** (ang. Voice over Internet)
- Komunikacja satelitarna
- Sprzęt nawigacyjny
- Modemy
- Poczta głosowa
- Automatyczne sekretarki
- Systemy wideokonferencjne
- Cyfrowe kamery
- Sonary
- Radary
- Sterowanie napędami
- Systemy zapobiegania kolizji pojazdów
- Systemy bezpieczeństwa w komunikacji
- Analiza sygnałów sejsmicznych
- Realizacja nagrań fonicznych
- Synteza dźwięków muzycznych i efektów dźwiękowych
- Usuwanie szumu
- Ultradźwiękowe systemy diagnostyki medycznej

2. Przykłady zastosowań procesorów sygnałowych - zastosowania DSPs w systemach mikroprocesorowych

- Filtracja cyfrowa
- Modulacja i demodulacja
- Systemy korekcji błędów: odporne na błędy kodowanie i dekodowanie sygnałów
- Kompresja sygnałów fonicznych
- Systemy studyjne przetwarzania sygnałów fonicznych:
 - systemy dźwięku przestrzennego np. DOLBY Surround
 - redukcja szumu
 - regulacja barwy dźwięku – equalizer
 - zmiana częstotliwości próbkowania
 - usuwanie echa
- Synteza sygnałów: mowy i muzyki
- Rozpoznawanie mowy

3. Omówienie cech charakterystycznych DSPs w porównaniu z GPs

- Magistrale danych i architektura *jednostki arytmetyczno-logicznej* (**ALU**
ang. Arithmetic Logic Unit)
- Zbiór instrukcji procesora
- Architektura pamięci
- Specjalizowane tryby adresowania pamięci
- Sterowanie wykonaniem programu
- Specjalizowane urządzenia peryferyjne **DSPs**

3.1. Magistrale danych i architektura ALU (1)

DSPs

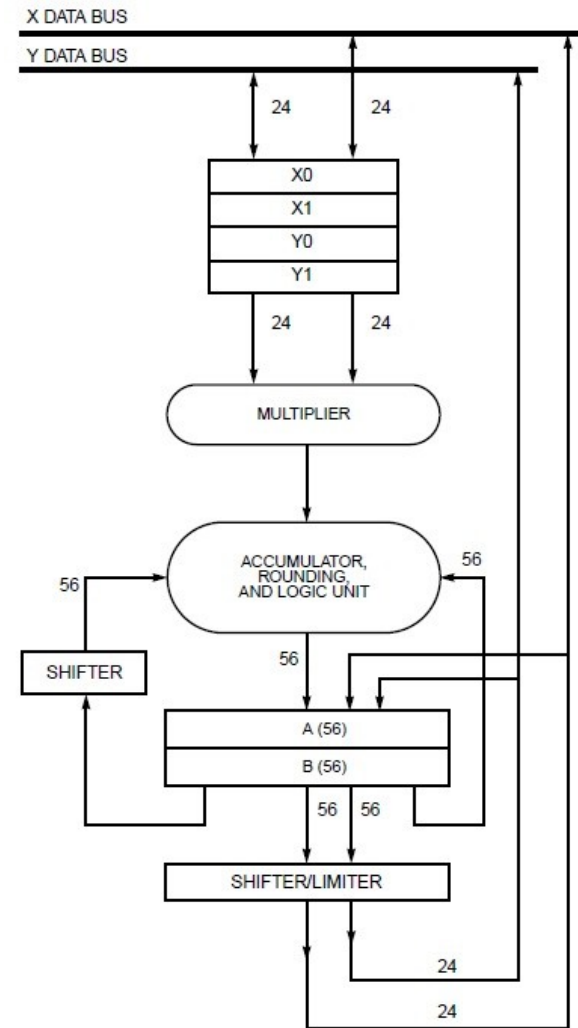
- Wykonywanie wszystkich głównych operacji arytmetycznych w jednym cyklu zegarowym
- Wsparcie sprzętowe dla zarządzania obliczeniami numerycznymi i ich dokładnością:
 - Rejestry skalujące
 - Linie opóźniające i bufor cyrkulacyjny
 - Bity kontrolne strumieni danych i rozkazów
 - Kontrola przepełnienia

GPSS

- Mnożenie często zajmuje więcej niż 1 cykl zegarowy
- Operacja przesunięcia często zajmuje więcej niż 1 cykl zegarowy
- Inne operacje (np. zaokrąglanie lub kontrola przepełnienia) zajmują zwykle kilka cykli zegarowych

3.1. Magistrale danych i architektura ALU (2) - architektura ALU przykładowego procesora sygnałowego

Architektura jednostki arytmetyczno-logicznej oraz magistrale danych typowego stałoprzecinkowego procesora sygnałowego – fragment schematu blokowego 24 bitowego **DSPs** z rodziny Motorola DSP560xx



3.2. Zbiór instrukcji procesora (1)

DSPs

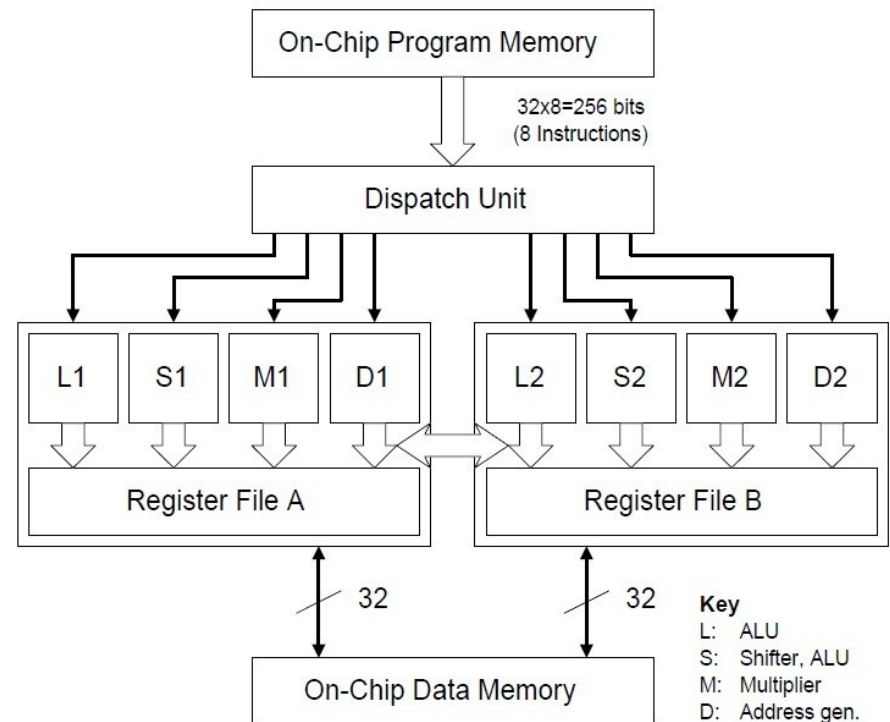
- Złożone, wyspecjalizowane instrukcje
- Wielokrotne operacje przypadające na jedną instrukcję (np. przy wykorzystaniu architektury z *bardzo długim słowem instrukcji* (VLIW ang. Very Long Instruction Word))

GPPs

- Instrukcje uniwersalne
- Typowo jedna operacja przypadająca na jedną instrukcję

3.2. Zbiór instrukcji procesora (2) – architektura VLIW

Mechanizmy zwiększania równoległości wykonywania instrukcji procesora, oparty na architekturze RISC (zredukowana liczba rozkazów i trybów adresowania, ograniczona komunikacja pomiędzy pamięcią a procesorem, zwiększona liczba rejestrów np. 32, 192 itp). Instrukcje są dzielone na części a następnie pakowane ponownie w duże instrukcje z dodatkową informacją odnośnie jednostki, na której mają być wykonywane. W rezultacie uzyskuje się pojedynczy o dużych rozmiarach opcode.



3.3. Architektura pamięci (1)

DSPs

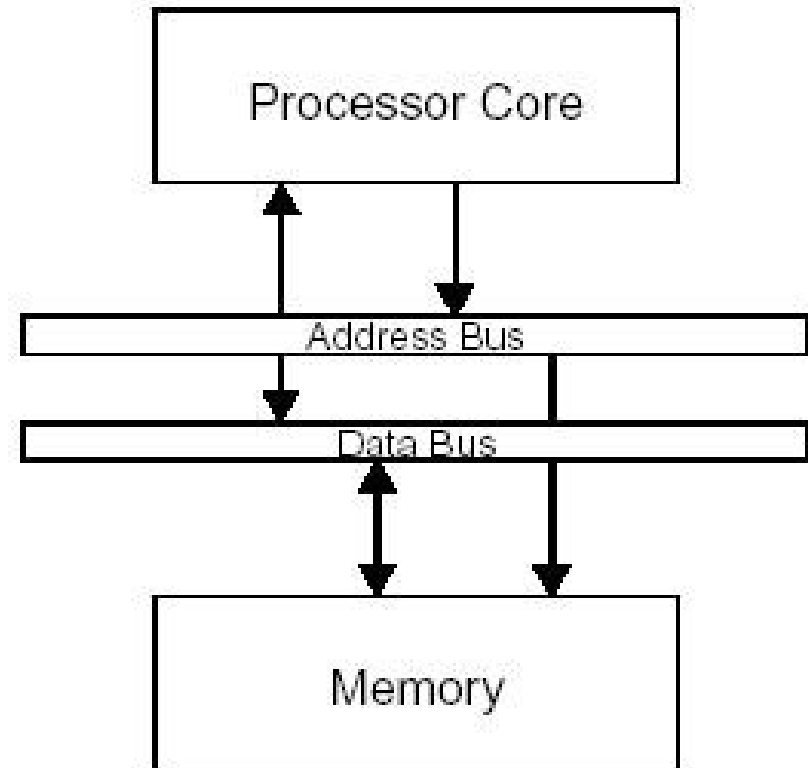
- Architektura Harvardzka
- Dostęp do 2-4 danych (komórek pamięci) na 1 cykl zegarowy
- Nie jest stosowana pamięć podręczna procesora (cache memory)

GPPs

- Architektura von Neumanna
- Typowo dostęp do 1 danej (komórki pamięci) na 1 cykl zegarowy
- Może być stosowana pamięć podręczna

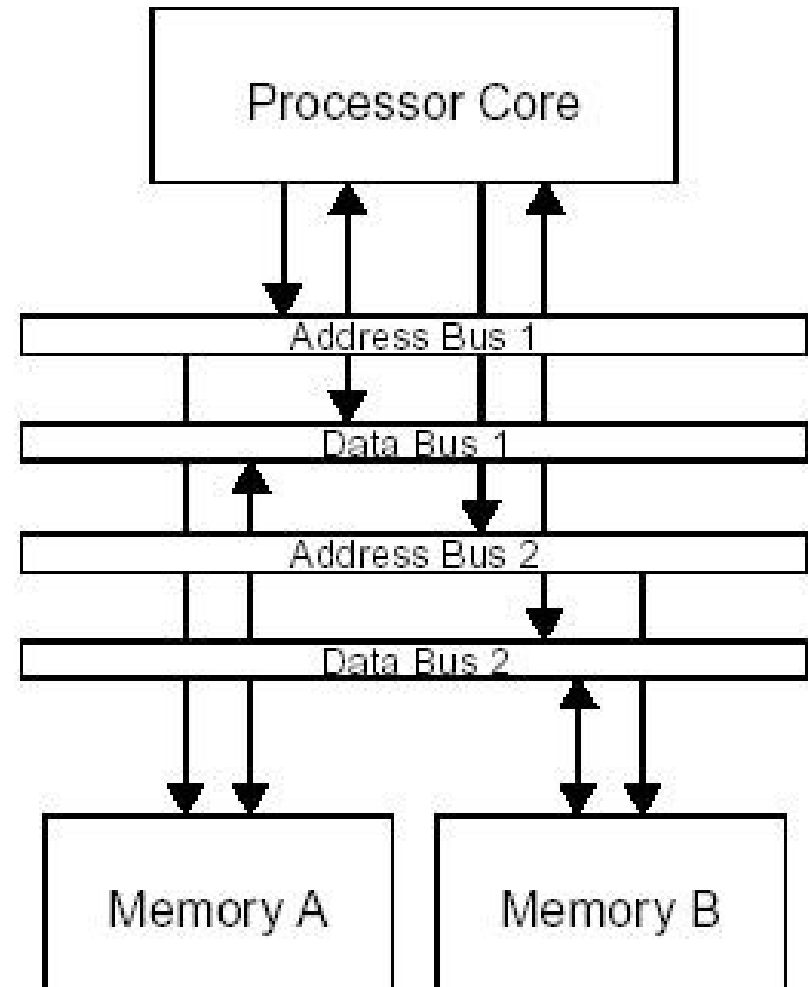
3.3. Architektura pamięci (2) - von Neumanna

Architektura pamięci von Neumana stosowana jest głównie w mikrokontrolerach oraz uniwersalnych mikroprocesorach. Ze względu na występowanie tylko jednej szyny danych, uaktualnienie zmiennych może odbywać się tylko wtedy, gdy nie jest pobierana z pamięci kolejna instrukcja programu – dane i program są zapisywane w tej samej pamięci.



3.3. Architektura pamięci (3) - Harvardzka

Harwardzka architektura pamięci stosowana jest głównie w **DSPs**'ach. Procesor sygnałowy może mieć jednocześnie dostęp do dwóch banków pamięci za pomocą dwóch niezależnych szyn danych. Dzięki temu może uaktualniać zmienne w trakcie pobierania kolejnych instrukcji programu.



3.4. Specjalizowane tryby adresowania pamięci

DSPs

- Dedykowane jednostki generowania adresu
- Specjalizowane tryby adresowania np.:
 - Automatyczna inkrementacja adresu
 - Adresowanie „modulo” (bufory cyrkulacyjne)
 - tzw. bit-reversed (FFT)
- Wsparcie bezpośredniego dostępu do danych

GPPs

- Zwykle nie występują wydzielone jednostki generowania adresu
- Uniwersalne tryby adresowania

3.5. Sterowanie wykonaniem programu

- Sprzętowe wsparcie dla „fast looping”
- „Fast interrupts” dla obsługi operacji wejścia/wyjścia
- Wsparcie trybu debugowania w czasie rzeczywistym

3.6. Specjalizowane urządzenia peryferyjne *DSPs*

- Porty „host” - do komunikacji z uniwersalnym komputerem np klasy PC sterującym pracą *DSPs*
- Bitowe porty We/Wy
- Zintegrowany kontroler *DMA*
- Generatory częstotliwości zegarowych
- Synchroniczne porty szeregowy
- Porty równoległe
- Timery
- Zintegrowane przetworniki *analogowo-cyfrowe A/D* i *cyfrowo analogowe D/A*

4. Klasyfikacja DSPs (1)

Podział procesorów sygnałowych według kryterium:

- Formatu arytmetycznego liczb (cyfrowej reprezentacji liczb)
 - Stałoprzecinkowe
 - Zmiennoprzecinkowe
 - Zmiennoprzecinkowe blokowe
- Rozmiaru słowa danych
 - 16 bitowe — typowe dla stałoprzecinkowych **DSPs**
 - 32 lub 64 bitowe — typowe dla zmiennoprzecinkowych **DSPs**
- Organizacji pamięci
- Możliwości pracy wieloprocessorowej

4. Klasyfikacja DSPs (2)

Podział procesorów sygnałowych według kryterium (cd):

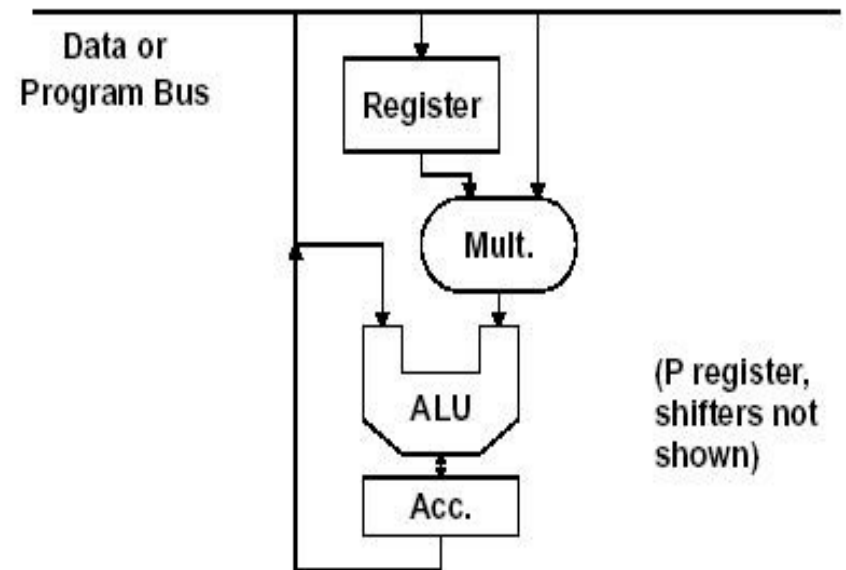
- Prędkości
 - Liczba milionów instrukcji na sekundę (**MIPS** ang. million of instruction per second)
 - Czas trwania wybranej instrukcji np: **MAC** - pomnóż i dodaj (zamiennie - liczba operacji na sekundę)
 - Czas trwania wybranego algorytmu np: **FFT**, filtracji **FIR** lub **IIR**
 - Czas wykonywania określonego programu testowego (**Benchmark program**)
- Zużycia energii
 - Wielkość napięcie zasilania
 - Występowanie trybu uśpienia oraz bezczynności
 - Możliwość programowania dzielników częstotliwości zegarów
 - Możliwość sterowania pracą **DSPs** za pomocą urządzeń peryferyjnych

5. Historia rozwoju DSPs

- Pierwsza generacja (***TI TMS32010***)
- Druga generacja (***Motorola DSP56001, AT&T DSP16A, Analog Dev. ADSP-2100, TI TMS320C50***)
- Trzecia generacja (***Motorola DSP56301, TI TMS320C541, TI TMS320C80, Motorola MC68356***)
- Czwarta generacja (***TI TMS320C6201, Intel Pentium MMX***)

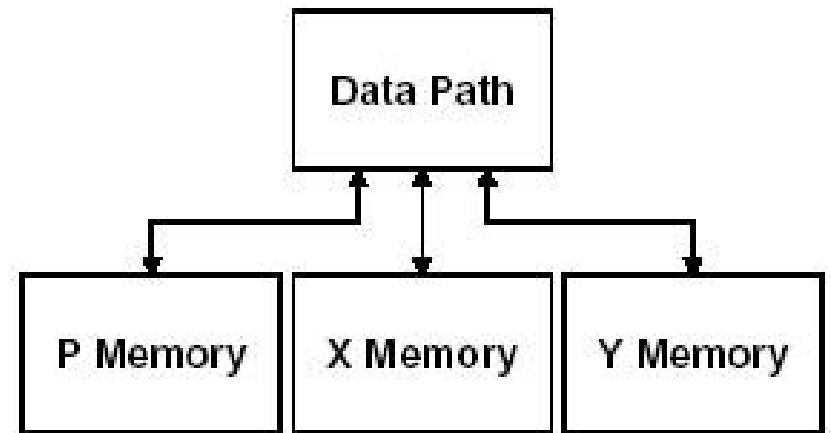
5.1. Pierwsza generacja DSPs (1982)

- Stałoprzecinkowe, 16 bitowe
- Harvardzka architektura pamięci
- Akumulator
- Zbiór wyspecjalizowanych instrukcji procesora
- Czas wykonania instrukcji **MAC**: 390 ns



5.2. Druga generacja DSPs (1987)

- 24 bitowe dane i instrukcje
- Harwardzka architektura pamięci
- 3 przestrzenie adresowe (P,X,Y)
- "Parallel move" przemieszczanie danych łącznie z wykonaniem instrukcji
- Pojedyncze i złożone instrukcje procesora
- "Hardware loops" – hardwareowe wsparcie wykonywania pętli
- Tryb adresowania modulo (bufory cykliczne)
- Bity kontrolne strumieni danych i rozkazów
- Czas wykonania instrukcji **MAC**: 75 ns



5.3. Trzecia generacja DSPs (1995)

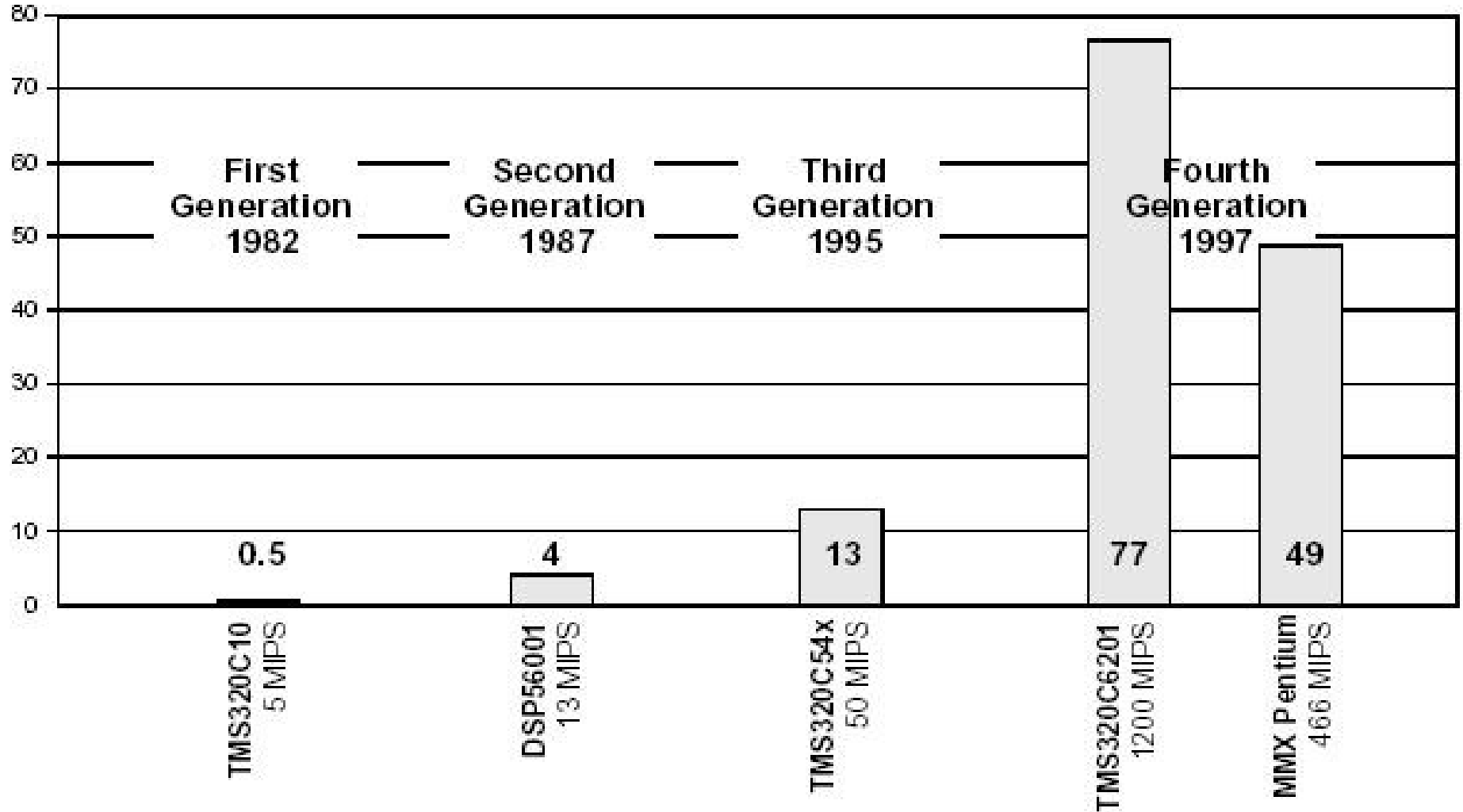
- Rozszerzenie dotychczasowej architektury **DSPs**
- Obniżenie napięcia zasilania do 3,0 lub 3,3 V — obniżenie wydzielanej mocy oraz zwiększenie szybkości działania procesora
- Zwiększenie pamięci wewnętrznej **DSPs**
- Nowe jednostki procesora służące do realizacji wyspecjalizowanych zadań np. w postaci dołączanych koprocessorów
- Bardziej wyrafinowane metody uruchamiania i debugowania programów:
JTAG, OnCE
- Budowa **DSPs** z wykorzystaniem architektury opartej o pojęcie rdzeni procesora: **DSPs cores**
- Wyposażenie **DSPs** w przetworniki **AC** i **CA**
- Czas wykonania instrukcji **MAC**: 20 ns

5.4. Czwarta generacja DSPs (1998)

- Znaczne zwiększenie szybkości zegarów procesora
- Obniżenie napięcia zasilania do 2,7 V
- Architektura **VLIW** - z *bardzo długim słowem instrukcji*
- Zwiększenie zużycia energii – kosztów eksploatacji urządzeń wyposażonych w **DSPs**
- Czas wykonania instrukcji **MAC**: 3 ns

5.5. Wykres rozwoju DSPs

BDTmark



6. Testy wydajności wybranych DSPs (1) *BDTmark*

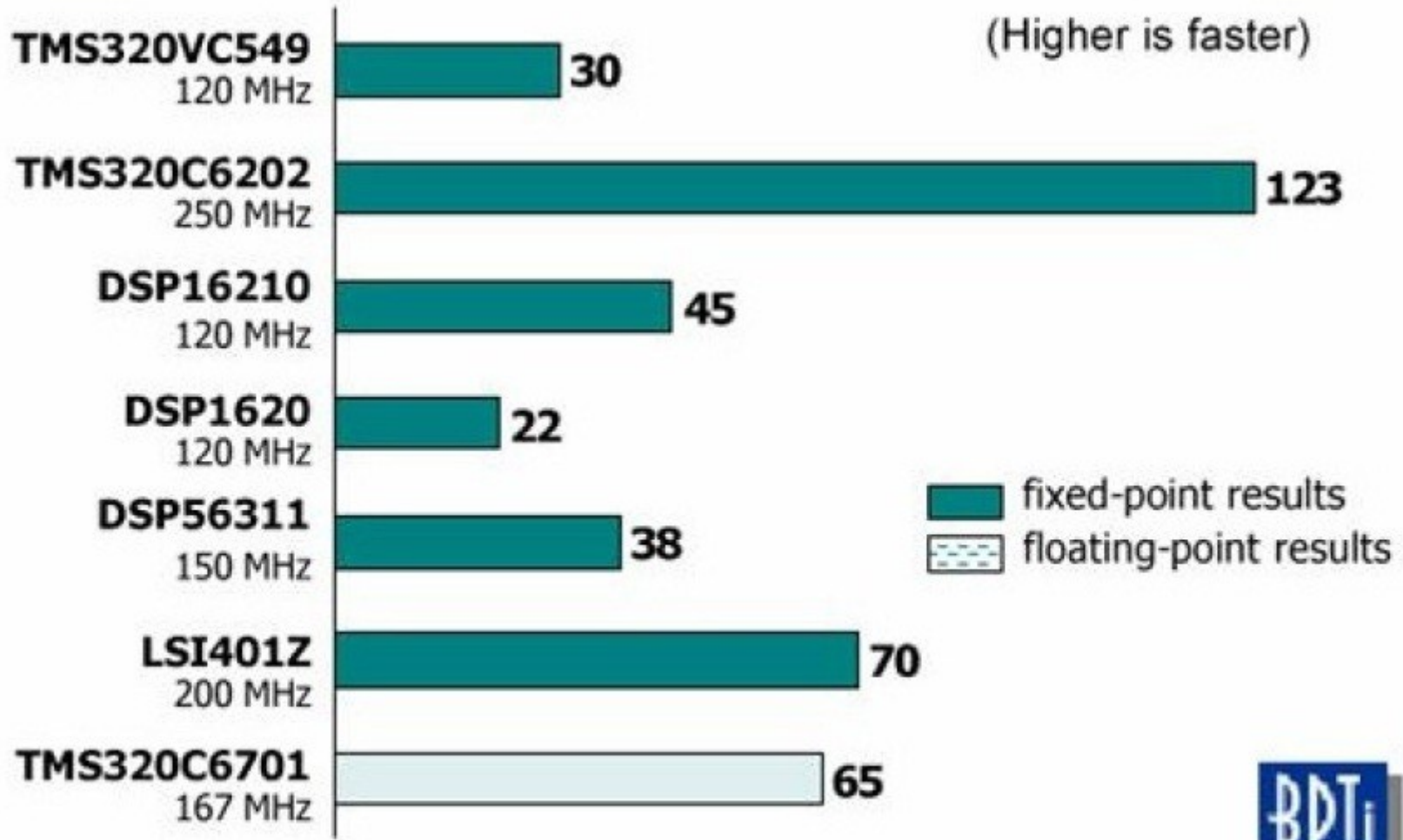
BDTmark jest najszerzej stosowanym testem wzorcowym wydajności DSPs. Został opracowany w Berkeley Design Technology Inc. Oparty jest na algorytmach istotnych dla typowych aplikacji **DSPs**, które zostały dobrane i „ręcznie” dopasowane do każdego procesora. W ramach testu **BDTmark** stosowane są między innymi algorytmy:

- filtracji **FIR** dla bloku danych rzeczywistych,
- filtracji **FIR** dla bloku danych zespolonych,
- filtracji **IIR**
- iloczynu skalarnego dwóch wektorów,
- sumy dwóch wektorów,
- poszukiwania maksymalnej wartości wektora,
- 256-punktowego algorytmu **FFT**

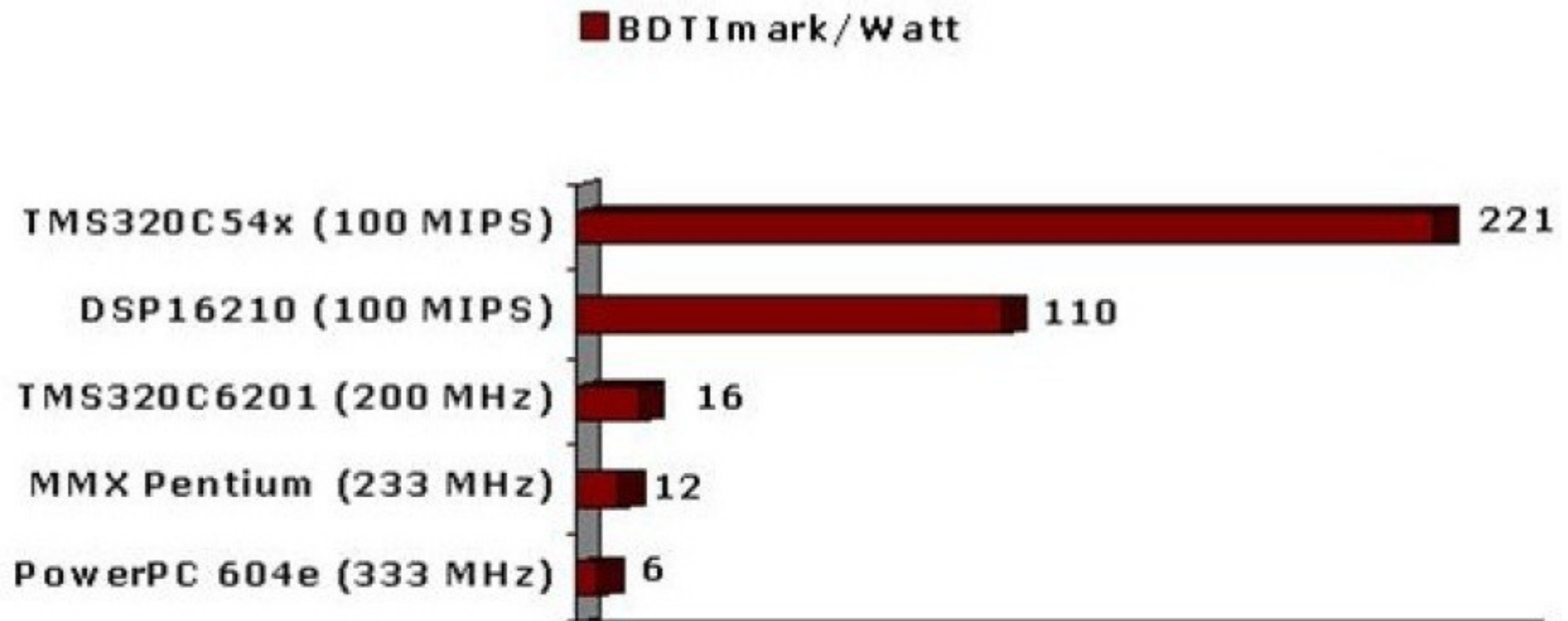
Do oceny stopnia zużycia energii oraz kosztów zakupu układu scalonego **DSPs** opracowano również testy **BDTmark / Wat** i **BDTmark / dolar**.

6. Testy wydajności wybranych DSPs (2)

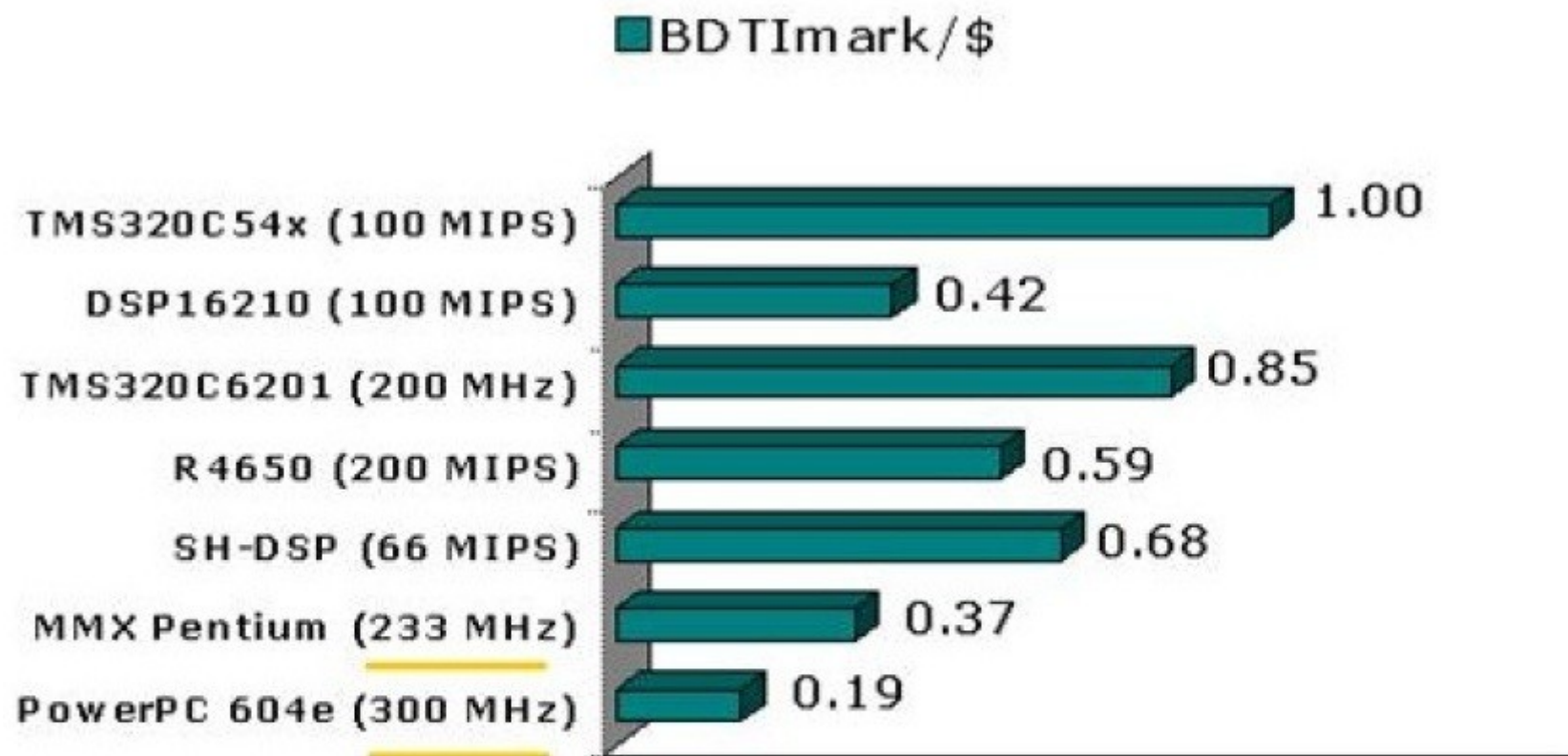
Example BDTImark results.



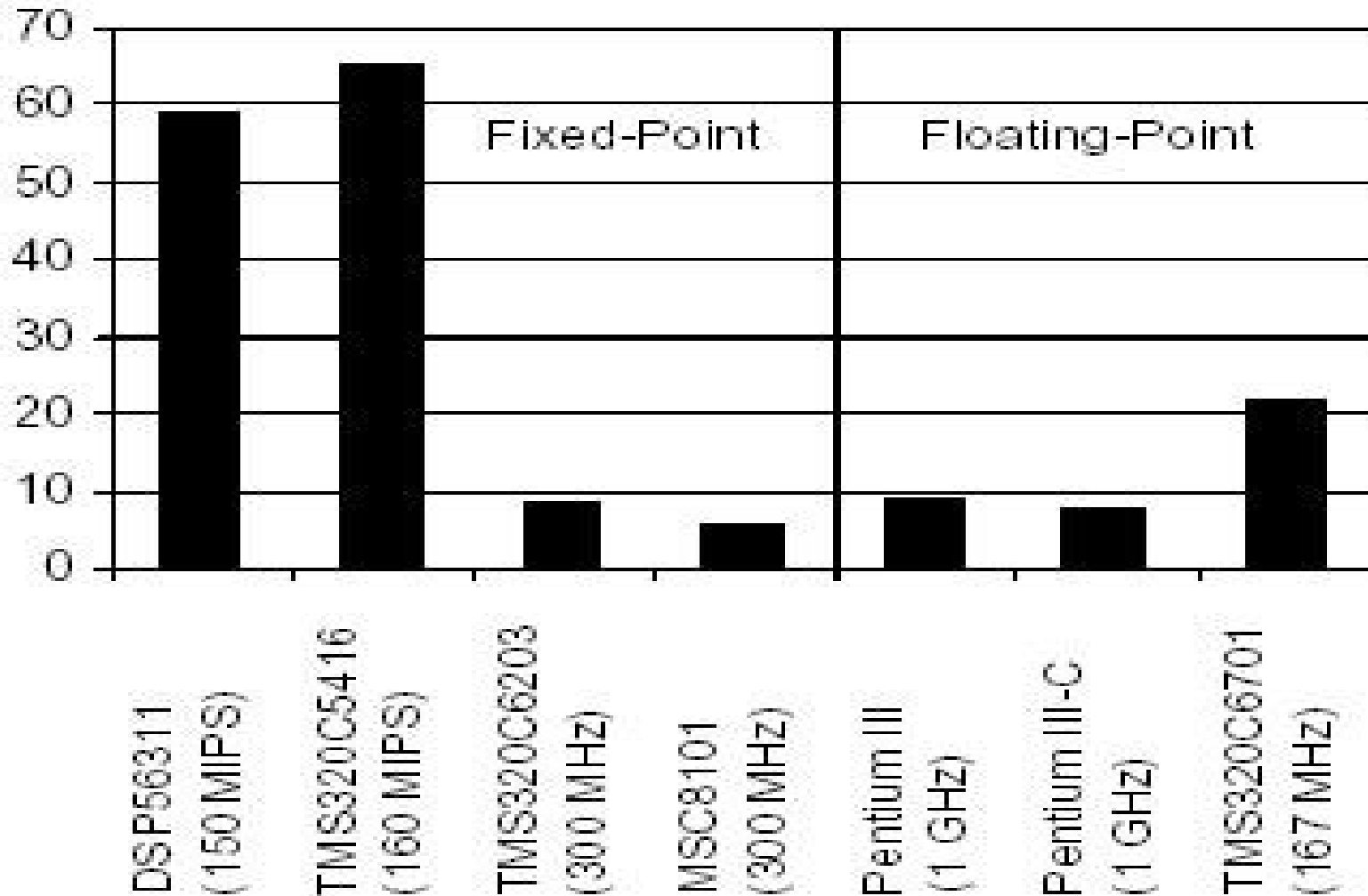
6. Testy wydajności wybranych DSPs (3)



6. Testy wydajności wybranych DSPs (4)



6. Testy wydajności wybranych DSPs (5)



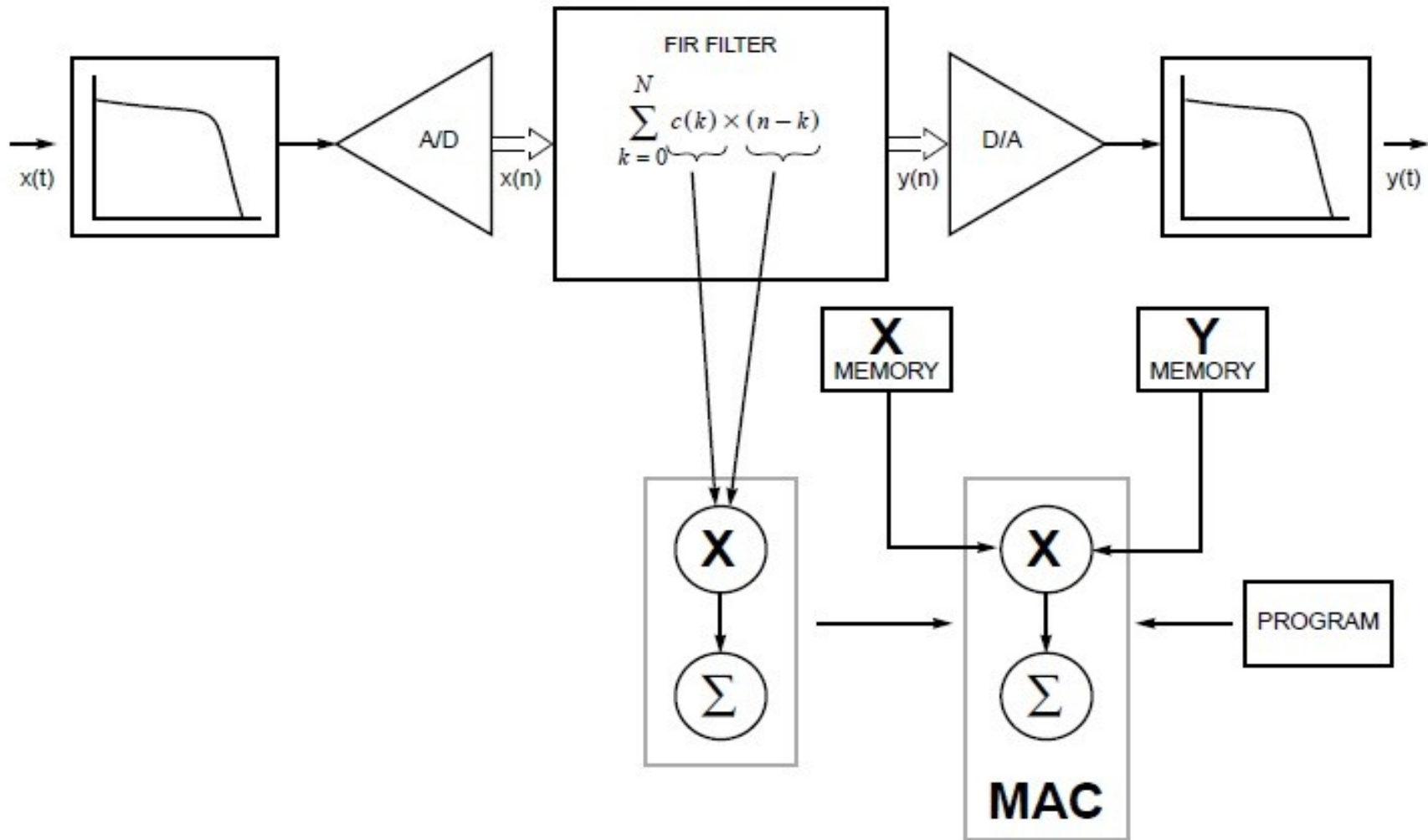
Czas w mikrosekundach obliczania
256 punktowej zespolonej **FFT**

7. Producenci procesorów sygnałowych

- Texas Instruments - USA
- Analog Devices - USA
- Motorola - USA

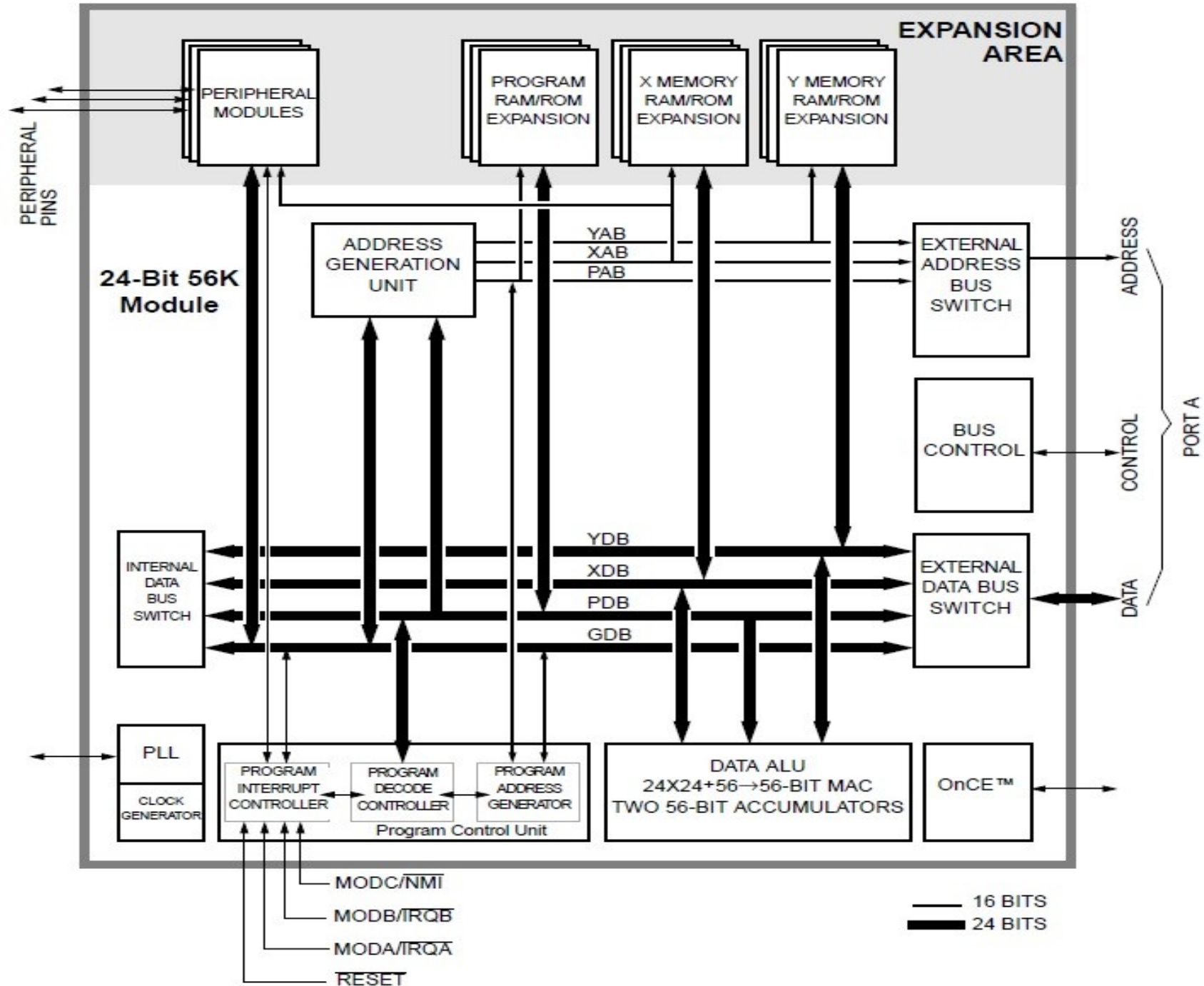
Procesory z wbudowaną jednostką **DSPs** oferowane są obecnie przez większość producentów uniwersalnych procesorów (dodatkowe rozkazy i mechanizmy)

8.1. Budowa DSPs - przykłady



Przykład zastosowania **DSPs** - filtracja liniowa **FIR**

8.2. Budowa DSPs – Motorola DSP560xx (1)



8.2. Budowa DSPs – Motorola DSP560xx (2)

Rodzina DSP56K procesorów opiera się na module standardowym jednostki centralnej. Poszczególne procesory różnią się innymi konfiguracjami pamięci i urządzeń peryferyjnych.

Główne składniki modułu jednostki centralnej:

- Magistrale danych
- Magistrale adresowe
- Jednostka arytmetyczno-logiczna danych (ang. Data Arithmetic Logic Unit - data ALU)
- Jednostka generatora adresu (ang. Address Generation Unit - AGU)
- Człon sterowania programem (ang. Program Control Unit - PCU)
- Port rozszerzenia pamięci (Port A)
- Wbudowany emulator (ONCE™)
- Układ zegara oparty o fazową pętlę sprzężenia zwrotnego (ang. Phase-Locked Loop - PLL)

8.2. Budowa DSPs – Motorola DSP560xx (3)

Magistrale danych

Moduł jednostki centralnej DSP56K zorganizowany jest wokół rejestrów trzech niezależnych jednostek wykonawczych: ALU, AGU i PCU. Przepływ danych pomiędzy jednostkami wykonawczymi odbywa się za pośrednictwem czterech dwukierunkowych 24-bitowych magistral:

- magistrali danych X (XDB),
- magistrali danych Y (YDB),
- magistrali danych programu (PDB),
- magistrali danych globalnych (GDB).

(Niektóre instrukcje łączą magistrale danych X oraz Y i traktują je jako jedną pojedynczą 48-bitową szynę danych)

Transfer danych pomiędzy ALU i pamięciami X lub Y odbywa się za pośrednictwem odpowiednio: XDB i YDB. Wszystkie inne transfery danych, takie jak np. operacje I / O z urządzeniami peryferyjnymi odbywają się przy wykorzystaniu GDB.

Pobieranie słów instrukcji odbywa się równolegle za pośrednictwem PDB.

8.2. Budowa DSPs – Motorola DSP560xx (4)

Magistrale danych cd.

Magistrale pozwalają na przesyłanie danych zarówno pomiędzy: rejestrem a rejestrem, rejestrem a pamięcią oraz pamięcią a rejestrem.

W tym samym cyklu instrukcji można przesłać maksymalnie dwa słowa 24 bitowe i jedno słowo 56 bitowe. Transfer danych pomiędzy magistralami odbywa się za pomocą wewnętrznego przełącznika magistral (ang. Internal Data Bus Switch - IDBS).

Magistrale adresowe

Adresy wewnętrznej pamięci danych X i Y są określane na dwóch jednokierunkowych 16-bitowych magistralach - szynie adresowej X (XAB) oraz szynie adresowej Y (YAB). Adresy pamięci programu są określone na dwukierunkowej magistrali adresowej programu (PAB).

Obszar pamięci zewnętrznej jest adresowany za pośrednictwem pojedynczej, jednokierunkowej 16-bitowej magistrali sterowanej za pomocą trójwejściowego multipleksera, który może wybrać jedna z pamięci XAB, YAB lub PAB.

W jednym cyklu instrukcji możliwy jest dostęp tylko do jednej komórki pamięci zewnętrznej.

8.2. Budowa DSPs – Motorola DSP560xx (5)

Magistrale adresowe cd.

Jeśli w trakcie wykonywania jednej instrukcji programu konieczny jest dostęp do jednej komórki pamięci zewnętrznej – nie nastąpi spowolnienie wykonywania programu. W przeciwnym razie jeśli jedna instrukcja potrzebuje dostępu do dwóch lub trzech danych umieszczonych w pamięci zewnętrznej, nastąpi opóźnienie wykonania instrukcji odpowiednio o jeden lub dwa cykle.

Wewnętrzny przełącznik magistral (IDBS)

Transfery pomiędzy magistralami odbywa się za pośrednictwem wewnętrznego przełącznika magistral. Jest on podobny do macierzy przełączającej i może połączyć dwie wewnętrzne magistrale bez wprowadzania typowego dla przetwarzania potokowego opóźnienia. Własność ta ułatwia programowanie.

8.2. Budowa DSPs – Motorola DSP560xx (6)

Jednostka manipulacji bitami

Jednostka manipulacji bitami jest fizycznie umieszczona w bloku IDBS ponieważ musi mieć dostęp do każdego obszaru pamięci. Jednostka manipulacji bitami wykonuje operacje manipulacji bitami na komórkach pamięci, rejestrach adresu, rejestrach sterujących i rejestrach danych za pośrednictwem magistral XDB, YDB i GDB.

Jednostka arytmetyczno-logiczna danych (data ALU)

ALU wykonuje wszystkie operacje logiczne i arytmetyczne na danych. Składa się z czterech 24 bitowych rejestrów wejściowych, dwóch 48 bitowych rejestrów akumulatora, dwóch 8 bitowych rejestrów rozszerzeń akumulatora, rejestru skalującego akumulatora, dwóch układów skalująco/ograniczających magistrali danych oraz jednostki równoległej dokonującej w jednym cyklu zegarowym niepotokowej operacji mnożenia i dodawania (ang. Multiply-Accumulator - MAC).

8.2. Budowa DSPs – Motorola DSP560xx (7)

Jednostka generowania adresu (AGU)

AGU wykonuje wszystkie operacje zapamiętania i obliczenia adresu potrzebne do pośredniego adresowania danych w pamięci. Działa równolegle z innymi zasobami procesora w celu zminimalizowania czasu generowania adresu. AGU ma dwie identyczne jednostki arytmetyczne adresu, które może generować dwa 16-bitowe adresy w każdym cyklu zegarowym. Każda z jednostek arytmetycznych może wykonać trzy typy operacji arytmetycznych: liniowa, modulo, reverse-carry.

Człon sterowania programem (PCU)

Człon sterowania programem wykonuje: pobieranie z wyprzedzeniem instrukcji, ich dekodowanie, sterowanie hardware'owo wspomaganymi pętlami, obsługę przerwań i wyjątków. Składa się z trzech modułów: generatora adresu programu, sterownika dekodera programu oraz sterownika przerwań programu. Zawiera 15-poziomą 32-bitową pamięć stosu systemowego oraz sześć bezpośrednio adresowalnych rejestrów: licznik programu (PC), adres pętli (LA), licznik pętli (LC), rejestr stanu (SR), rejestr trybu obsługi (OMR) i wskaźnik stosu (SP). 16-bitowy licznik programu (PC) może zaadresować 65536 komórek w pamięci programu.

8.2. Budowa DSPs – Motorola DSP560xx (8)

Człon sterowania programem (PCU) cd.

Układ posiada cztery piny zewnętrzne pozwalające na zmian trybu i rodzaju przerwań obsługiwanych przez sterownik przerwań programu. Piny Mode Select A/External Interrupt Request A (MODA/IRQA) oraz Mode Select B/External Interrupt Request B (MODB/IRQB) pozwalają na wybór trybu pracy układu i otrzymywania żądań przerwań od urządzeń zewnętrznych.

Pin Mode Select C/Non-Maskable Interrupt (MODC/NMI) pozwala na wybór kolejnego trybu pracy układu oraz wywołanie niemaskowanego przerwania. Pin RESET resetuje cały układ.

Port rozszerzenia pamięci (Port A)

Port A służy do łączenia za pomocą 24-bitowej szyny danych z szerokim wyborem pamięci i urządzeń zewnętrznych. Do urządzeń tych należą szybkie statyczne pamięci RAM, inne wolniejsze urządzenia pamiętające, inne DSPs i mikroprocesory (MPUs) pracujące w konfiguracjach MASTER/SLAVE. Taka różnorodność dopuszczalnych urządzeń zewnętrznych wynika z możliwości programowania parametrów zegara szyny rozszerzeń.

8.2. Budowa DSPs – Motorola DSP560xx (9)

Wbudowany emulator (ONCE™)

Układ wbudowanego emulatora (ONCE) pozwala użytkownikowi na współpracę z DSP56K i jego urządzeniami peryferyjnymi bez wpływania na bieżąco wykonywane programy. Dzięki niemu można przeglądać i zmieniać zawartość rejestrów, pamięci albo odczytywać lub zmieniać stan urządzeń peryferyjnych. Emulator pozwala na stosowanie wyszukanych metod debugowania programów w czasie rzeczywistym, a tym samym ułatwia uruchamianie skomplikowanych aplikacji.

Układ zegara oparty o fazową pętlę sprzężenia zwrotnego (ang. Phase-Locked Loop – PLL)

PLL pozwala DSPs użyć prawie każdego zewnętrznego zegara systemowego dla pracy z pełną prędkością przetwarzania. Jednocześnie zapewnia pełną synchronizację tego zegara z zsyntetyzowanym zegarem wewnętrznym.

Bibliografia

- Chassaing Rulph, *Digital Signal Processing and Applications with the C6713 and C6416 DSK*, Wiley-Interscience 2004.
- *DSP56000, 24-BIT DIGITAL SIGNAL PROCESSOR FAMILY MANUAL*, Motorola, Inc. Semiconductor Products Sector DSP Division 6501 William Cannon Drive, West Austin, Texas 78735-8598, 1995.
- Phil Lapsley, Jeff Bier, Amit Shoham, and Edward A. Lee, *DSP Processor Fundamentals: Architectures and Features*, Berkeley, California: Berkeley Design Technology, Inc., 1996.
- Jennifer Eyre and Jeff Bier, Berkeley Design Technology, The Evolution of DSP Processors, A BDT I White Paper, Berkeley Design Technology, Inc. 2000.
- Choosing a DSP Processor, A BDT I White Paper, Berkeley Design Technology, Inc. 2000, <http://www.BDTI.com>.
- <http://www.eg3.com/dsp>