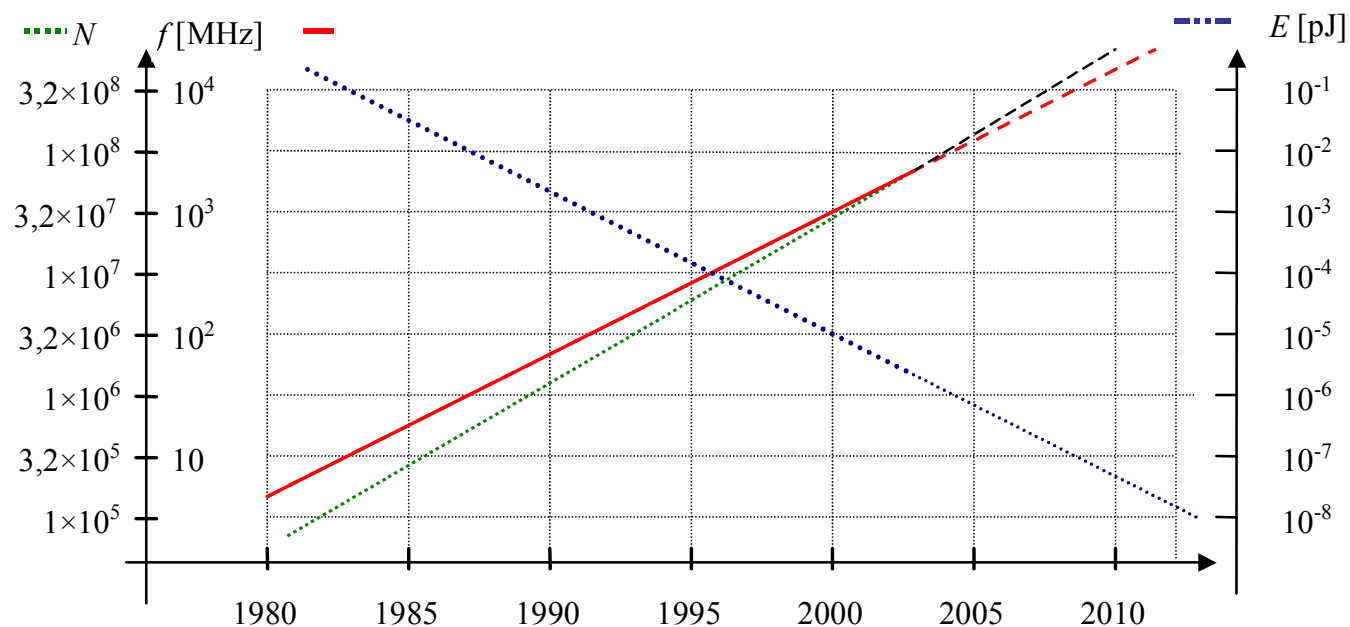


Komputery współczesne – ewolucja architektury Pentium™

	Procesor	N_{TRANZ}	f_{max}	MIPS	adres	nowe cechy
1976	8085A	6,2 K	2MHz		64KB	
1978	8086 / 8088	29 K	10 MHz	0,8	1 MB	(koprocesor num. 8087)
1982	80286	134 K	12,5 MHz	2,7	16 MB	<i>tryb wirtualny 24b</i>
1985	386 DX / SX	375 K	33 MHz	6	4 GB	<i>tryb wirtualny 32b</i>
1989	486 DX / SX	1,2 M	66 MHz	20		<i>cache 8 KB, FPU</i>
1992	Pentium (P5)	3,1 M	100 MHz	100		<i>cache 8+8 KB</i>
1993	Pentium MMX	3,1 M	100 MHz	100		<i>SIMD – MMX</i>
1995	Pentium Pro (P6)	5,5 M	333 MHz	440	64 GB	<i>cache II 256/512 KB</i>
1997	Pentium II	7 M	450 MHz	466		<i>SIMD – SSE</i>
1999	Pentium III	8,2 M	450 MHz	1000		<i>data flow ALU</i>
1999	P III Xenon	28 M	733 MHz			<i>SIMD – SSE2</i>
2000	Pentium 4 (NB)	42 M	4,33 GHz	1500		<i>CISC shell, RISC core</i>
2001	Xenon (NB)					<i>cache I: 12 Kμop+8 KB (data)</i> <i>cache II:256 KB, cache III:2 GB</i>
2005	P Core2	>100 M				<i>2 rdzenie wykonawcze</i>
	core i5					
2009	core i7	500 M				<i>wiele rdzeni</i>
						<i>GP GPU (ang. General Purpose Graphic Processing Unit)</i>

Obserwowane tempo rozwoju technologii półprzewodnikowych



Liczba tranzystorów w układzie scalonym N , częstotliwość taktowania mikroprocesorów f [MHz] wydatek energetyczny na działanie logiczne E [pJ]

Prawa Gordona Moore'a

Tempo rozwoju technologii VLSI – obserwacja empiryczna:

„Co 18 miesięcy podwaja się gęstość upakowania układów VLSI (pojemność pamięci półprzewodnikowych) oraz osiągalna szybkość przełączania”

Faktyczna zależność jest wykładnicza (t – rok prognozy): $V(t) = V_0 \cdot 2^{(t-t_0)/a}$

V_0	Parametr $t_0=$	1980	2000	2010	[...]	a [lat]
N_0	Liczba tranzystorów w IC	$4,75 \cdot 10^4$	$3 \cdot 10^7$	$75 \cdot 10^7$	[]	2,15
C_0	Pojemność DRAM	2^{-12}	1	25	[Gb]	2,15
f_0	Częstotliwość przełączania	$4,77 \cdot 10^{-3}$	1	15	[GHz]	2,59
E_0	Energia 1 przełączenia	0,5	10^{-5}	$45 \cdot 10^{-9}$	[pJ]	-1,28
G_0	Przepustowość procesora	4	500	$6 \cdot 10^3$	[MIPS]	2,81
H_0	Przepustowość komputera	0,5	300	$6 \cdot 10^3$	[GIPS]	2,25

Obserwacja: *Koszty wdrożenia technologii rosną szybciej niż zyski ze sprzedaży*

Hipoteza (II prawo) G.Moore'a – „Wzrost kosztu linii produkcyjnej układów scalonych dla nowych technologii spowoduje utratę opłacalności wytwarzania nowych układów.

Obserwacja: *Przestało już obowiązywać ...*

Hipoteza Landauera

Rozwój technologii VLSI i prognoza

Rok	Grubość warstwy	Liczba warstw	Zasilanie	Częstotliwość
1985	1,6 μm	1	±5,0 V, 12,0 V	12 MHz
1990	1,0 μm	2	5,0 V	50 MHz
1995	0,5 μm	4	3,3 V	150 MHz
2000	0,15 μm	6	1,8 V	1 GHz
2005	0,08 μm	8	1,3 V	4 GHz
2010	0,06 μm	8	1,1 V	15 GHz

stała Landauera – minimalna energia przełączenia stanu atomowego

$$e = k T \ln 2 = 0,956751 \cdot 10^{-11} \cdot T \text{ [pJ]} \quad (k = 1,3803 \cdot 10^{-11} \text{ [pJ/deg]} - \text{stała Boltzmannna}),$$

W typowej dla US temperaturze otoczenia $T \cong 315^\circ\text{K}$ (42°C) jest $e \cong 3 \cdot 10^{-9} \text{ [pJ]}$

Hipoteza Landauera: *Granicznym etapem rozwoju technologii półprzewodnikowych jest obniżenie energii przełączania do poziomu minimalnego*

$$E(t) = E(t_0) \cdot 2^{-1,28(t-t_0)} \geq e(T) \Rightarrow t \cong 2013$$

Rozwój: **przetwarzanie równoległe**

instrukcje, procesy, procesory wielordzeniowe

Przetwarzanie równoległe i współbieżne

Równoległe przetwarzanie instrukcji jednego programu

W sekwencji działań występują zestawienia instrukcji, których kolejność wykonania może zostać zmieniona bez wpływu na algorytm; taka niezależność jest dość częsta w zestawieniu 2 instrukcji, jest też znacząca w zestawach 3 instrukcji:

- przetwarzanie superskalarne
- koncepcja VLIW (ang. *Very Large Instruction Word*) – wspólne kodowanie kilku kolejnych działań na poziomie kompilacji
- przetwarzanie w trybie *data flow*

Równoległe przetwarzanie wątków jednego procesu

wieloprocесory

- osiągalny stopień zrównoleglenia – prawo Amdahl'a (szczególny przypadek prawa malejących przychodów)

Równoległe przetwarzanie procesów

systemy wieloprocесorowe

- osiągalna obciążalność systemu (ang. *workload*) – prawo Gustafsona
- skrócenie zwłoki wykonania zadania (ang. *latency*) (czas odpowiedzi)

Równoległe przetwarzanie instrukcji jednego programu

Problemy w przetwarzaniu superskalarnym (ang. *Instruction Level Parallelism*):

- zależności danych
 - RAW (ang. *true data dependency, flow dependency*),
 - WAR (ang. *output dependency*),
 - WAW (ang. *antidependency*)
- zależność proceduralna (ang. *procedural dependency*)
 - rozgałęzienia (ang. *branches*)
 - procedury i powroty (ang. *calls and returns*),
- konflikt zasobów (ang. *resource conflicts*)
 - dostęp do pamięci (ang. *memory access*)
 - instrukcje czasochłonne (ang. *time-consuming instructions*)

bufor zmiany kolejności (ang. *re-order buffer, ROB*)

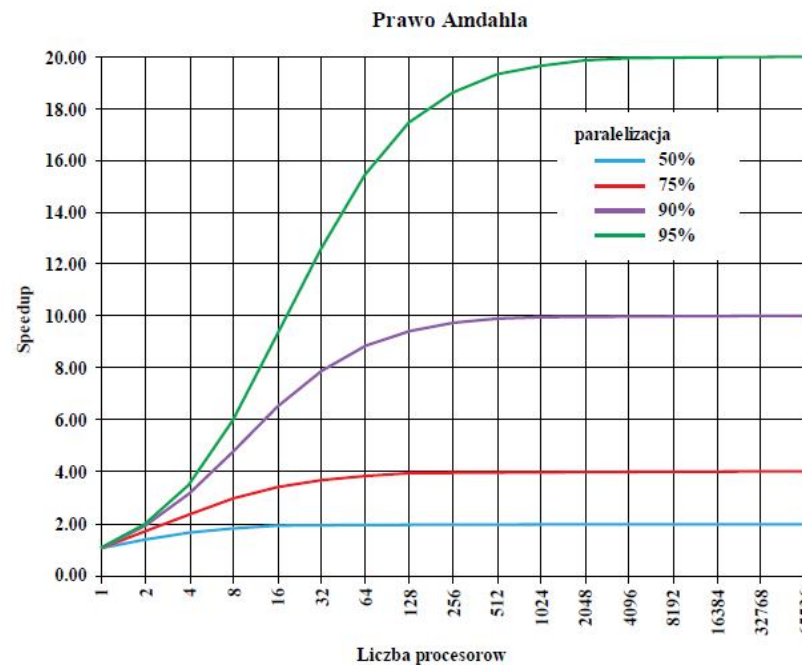
- stan mikrooperacji (w kolejce – wykonywana – zakończona)
- adres instrukcji, która wytworzyła mikrooperację (instr. prostą)
- kod mikrooperacji
- rejestr zastępczy (sprzętowy, jeden ze 128) dla architekturnego

Prawo Amdahla (1967)

Przyśpieszenie przetwarzania (ang. speed-up) jest ograniczone przez część algorytmu, której nie można wykonać równoległe z innymi jego częściami

$$S_{latency} = \frac{T(1-p) + Tp}{T(1-p) + Tpn^{-1}} = \frac{1}{1-p + \frac{p}{n}}$$

$1-p$ – udział części sekwencyjnej, n – liczba procesorów/zasobów zrównoleglonych



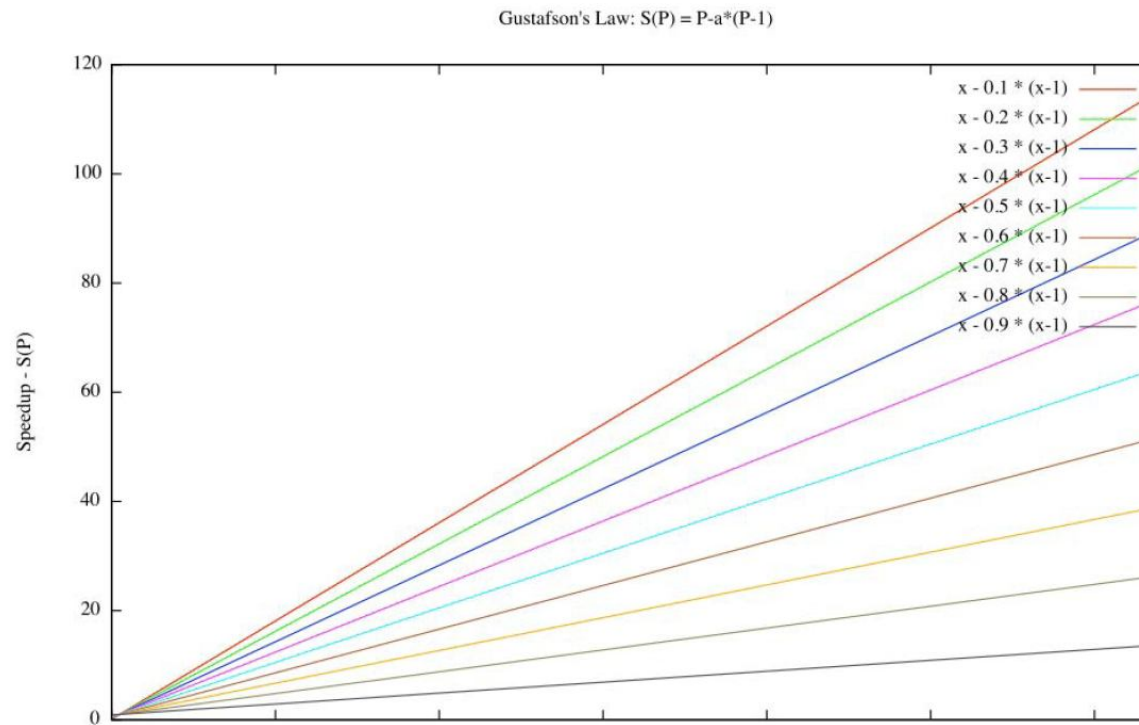
Prawo Gustafsona-Barsisa (1988)

Obciążenie robocze (ang. *workload*) po ulepszeniu w skali n mamy $W(n)=(1-p)W+npW$

Przyśpieszenie wykonania (skrócenie zwłoki w wykonaniu) zadania w ustalonym czasie wykonania (at fixed execution time), spodziewane po ulepszeniu zasobów w skali n .

$$S(n) = TW(n) / TW = W(n) / W = 1 - p + pn$$

- n – przyśpieszenie (*speed-up in latency*) wykonania części zadania z użyciem ulepszonych zasobów;
- p – udział części ulepszanej w nakładzie czasu pracy całego zadania (przed ulepszeniem);



Porównanie

Prawo Amdahla: Samochód pokonujący dystans 60 km pokonał w ciągu 1h połowę dystansu jadąc z prędkością 30 km/h. Jest niemożliwe, by uzyskał średnią 90 km/h na całej trasie.

Prawo Gustafsona: Samochód jakiś czas jechał z prędkością < 90 km/h. Jeśli wystarczy czasu i drogi może uzyskać bliską jej średnią niezależnie od czasu i odległości już przebytej.

Prawo Amdahla – perspektywa przyśpieszenia jednego zadania

- wymagania obliczeniowe pozostają stałe mimo wzrostu mocy obliczeniowej (analiza tych samych danych zajmie mniej czasu przy wzroście mocy obliczeniowej).
- wpływ wielu rdzeni na szybkość postawienia systemu operacyjnego; jeśli zrównoleglone operacje są wewnętrznie sekwencyjne, to dalsze zrównoleglanie nie da efektu
- nie uwzględnia bariery przepustowości pamięci oraz we/wy,
- zrównoleglanie algorytmu o złożoności $> O(N)$ jest nieefektywne (Snyder: zrównoleglenie części algorytmu o złożoności $O(N^3)$ powoduje wzrost jego rozmiaru o ok. 26%)

Prawo Gustafsona – perspektywa lepszego użycia zasobów w ustalonym czasie.

- wzrost mocy obliczeniowej umożliwi dokładniejszą analizę lecz nie większą skalę obliczeń.
- wzrost mocy spowoduje oczekiwanie wzrostu możliwości systemu, nawet kosztem jego dłuższej inicjalizacji systemu
- zrównoleglenie zadań wykonywanych na ograniczonym zbiorze danych (np. populacja ludności Ziemi) jest mało skuteczne w kontekście użycia zwiększonych zasobów,
- ograniczenia wykonania części sekwencyjnej mogą być zrekompensowane

Taksonomia Flynn'a – klasy przetwarzania?

SISD – Single Instruction Single Data

- procesor pojedynczy

SIMD – Single Instruction Multiple Data

- procesor wektorowy (ang. *vector processor*)
- procesor macierzowy (ang. *array processor*)

MISD – Multiple Instruction Single Data

- procesor wektorowy

MIMD – Multiple Instruction Multiple Data

- pamięć współdzielona (ang. *shared memory*), wieloprocесory (ang. *tightly coupled*)
 - multiprocesor symetryczny (ang. *symmetric multiprocessor, SMP*)
 - pamięć niejednolita (ang. *non uniform memory access, NUMA*)
- pamięć rozłożona (ang. *distributed memory*), klastry (ang. *clusters*)
- multikomputery z przesyłaniem wiadomości (ang. *message passing*).

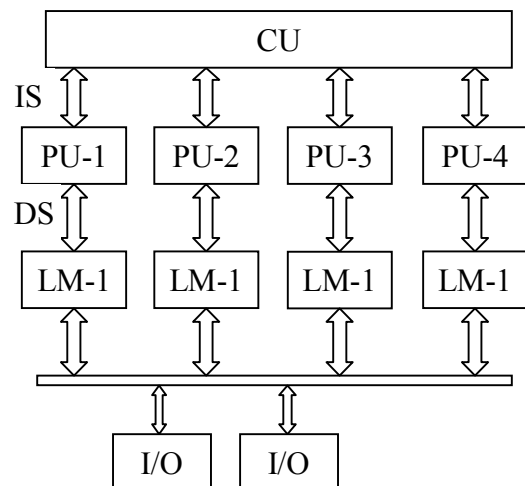
CP – Concurrent Processing (przetwarzanie współbieżne)

MP – Massive Parallelism (fizyczne zrównoleglenie) – wiele procesorów

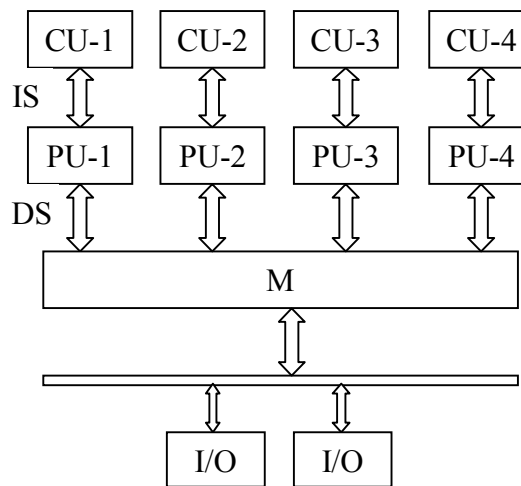
multicore – wiele jednakowych rdzeni w jednym układzie scalonym

many-core – rdzenie różne w jednym układzie

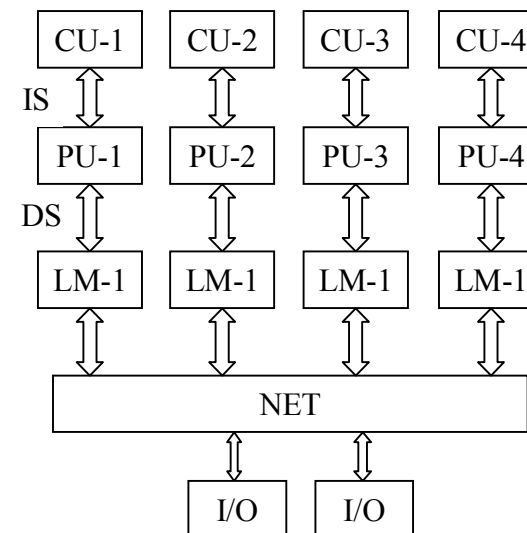
Architektura systemów równoległych



SIMD – pamięć rozłożona



MIMD – pamięć współdzielona



MIMD – pamięć rozłożona

CU – jednostka sterująca (ang. *Control Unit*)

PU – jednostka przetwarzająca (ang. *Processing Unit*)

LM – pamięć lokalna (ang. *Local Memory*)

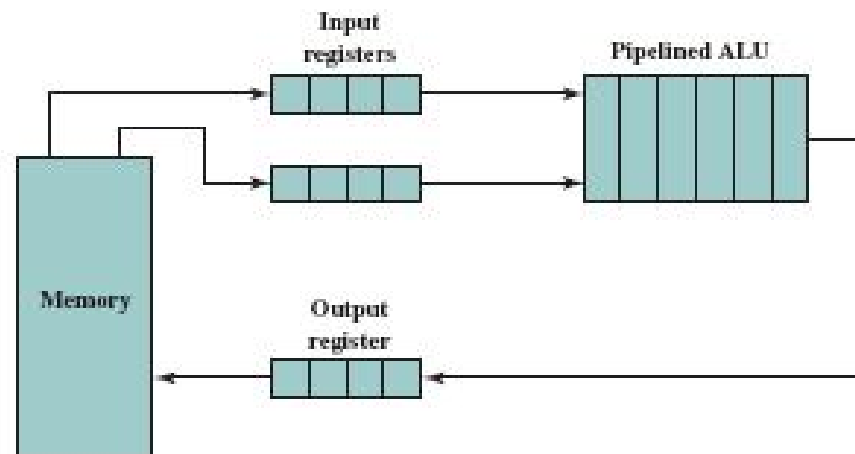
IS – strumień instrukcji (ang. *Instruction Stream*)

DS – strumień danych (ang. *Data Stream*)

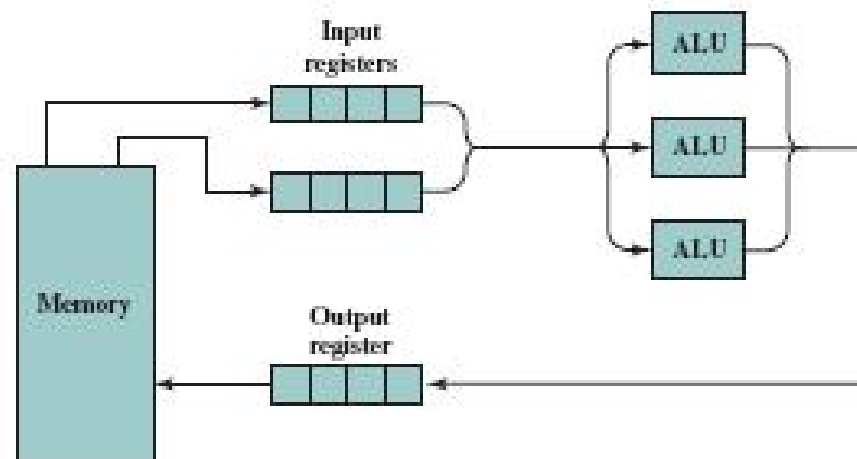
NET – sieć połączeń (ang. *Interconnection Network*)

SIMD - przetwarzanie wektorowe

- przetwarzanie potokowe (*pipelining*)



- przetwarzanie *data flow*



Tryby przetwarzania wielowątkowego

skalarne

z przełączaniem wątków (ang. *scalar multithreading*)

z przełączaniem bloków wątków (ang. *scalar multithreading*)

superskalarne

wielowątkowość z przeplotem (ang. *interleaved/fine grain multithreading*)

- w każdym cyklu przełączanie wątków, blokada wątku nie blokuje innych

wielowątkowość blokowa (ang. *blocked / coarse grain multithreading*)

- przełączanie wątków w razie blokady aktualnie wykonywanego

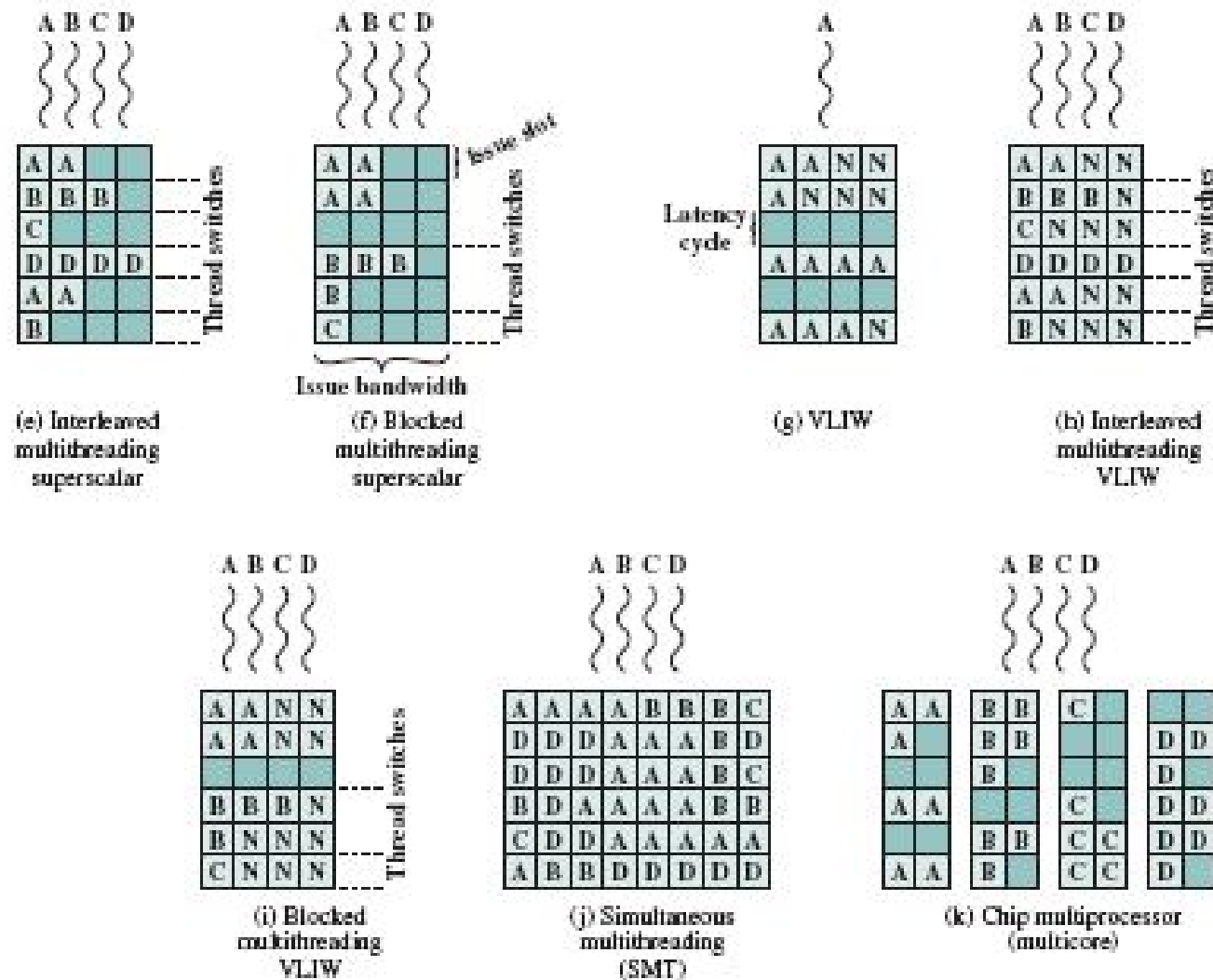
wielowątkowość współbieżna (ang. *simultaneous multithreading, SMT*)

- wątki realizowane współbieżnie w trybie superskalarnym

wieloprocusowość wieloprocusorowa (ang. *chip multiprocessing*)

- wątki realizowane współbieżnie przez różne procesory komputera

Przetwarzanie wielowątkowe



Wieloprocusowość i multiprocesor

Multiprocessing to użycie ≥ 2 CPU w pojedynczym systemie komputerowym wraz ze zdolnością systemu do wspierania >1 CPU i zdolnością alokacji zadań:

- wiele rdzeni w jednej kostce (ang. *chip multiprocessor*, CMP)
- wiele kostek (ang. *dies*) w jednym opakowaniu (ang. *chip package*)
- wiele zestawów (ang. *massively parallel*) w systemie.

multitasking – wykonywanie wielu zadań przez 1 procesor z podziałem czasu (ang. *time-sharing system*) (także *multiprogramming*)

multiprocessing – równoległe wykonywanie wielu procesów na wielu procesorach

parallel processing – użycie przez 1 proces (zadanie) wielu procesorów równocześnie

multiprocesor (ang. *tightly coupled system*) - system komputerowy zawierający wiele procesorów współdzielących pamięć (ang. *shared memory*) i peryferia (lub ich część)

W taksonomii Flynna multiprocesory są klasy MIMD

symetryczna wieloprocusorowość (ang. *symmetric multiprocessing*)

procesor wielordzeniowy – wieloprocusowość (ang. *multiprocessing*) w pojedynczym opakowaniu (ang. *package*).

Symetryczny multiprocesor

symetryczna wieloprocessorowość (ang. *symmetric multiprocessing*) – realizowana była za pomocą osobnych CPU, zarządzanie dobrze oprogramowane

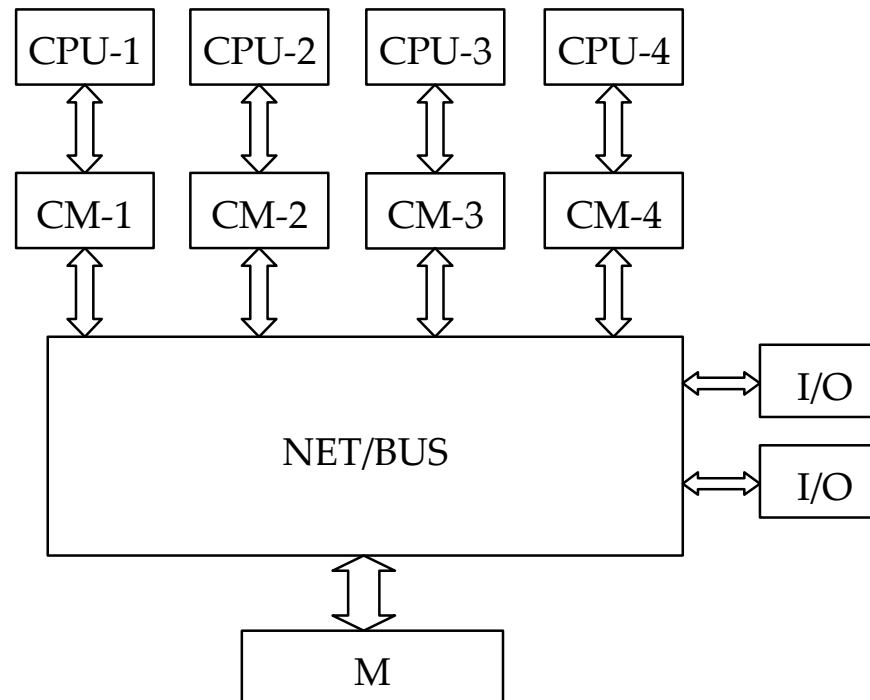
właściwości

1. zawiera co najmniej 2 podobne procesory o porównywalnym potencjale
2. procesory współdzielą pamięć i są połączone magistralą lub winny sposób zapewniający taki sam czas dostępu do pamięci
3. procesory współdzielą interfejsy we/wy albo przez ten sam kanał albo przez różne kanały do tych samych urządzeń
4. procesory wykonują takie same instrukcje
5. komputer jest zarządzany przez jeden zintegrowany system operacyjny, który zapewnia współdziałanie na poziomie procesów, zadań, plików i danych

potencjalne zalety

- a) większa wydajność, jeśli program można zrównoleglić
- b) większa dostępność i ftc – uszkodzenie jednego procesora zmniejsza wydajność
- c) dodanie procesora pozwala zwiększyć wydajność (niewiele – prawo Pollack'a)
- d) skalowalność architektury

Symetryczny multiprocesor – architektura



- komunikacja procesorów przez pamięć lub przez indywidualne łącza
- organizacja pamięci umożliwia fizyczną separację bloków i współbieżny dostęp
- procesory mogą dysponować lokalnymi pamięciami oprócz wspólnej głównej oraz dysponować własnymi urządzeniami we/wy
- konieczny rozbudowany sterownik przerwań (APIC)

Procesor wielordzeniowy

Procesor wielordzeniowy (ang. *multi-core processor*) element obliczeniowy z dwoma lub większą liczbą niezależnych jednostek przetwarzających - rdzeni (ang. *cores*).

Procesory wielordzeniowe są egzemplifikacją architektury SMP. Dodatkowo:

- użycie sprawdzonych projektów procesorów jako rdzeni, bez zmian architektury znacznie redukuje ryzyko błędów projektowych
- istotną motywacją tworzenia procesorów wielordzeniowych jest ograniczenie zysku przepustowości przez wzrost częstotliwości taktowania, z uwagi na:
 1. "mur" pamięci (ang. *memory wall*) – rosnąca luka (gap) między szybkością procesora i pamięci, co powoduje zwiększanie pojemności *cache* dopóki przepustowość pamięci (ang. *bottleneck*) nie staje się barierą wydajności,
 2. "mur" współbieżności instrukcji (ang. *instruction level parallelism wall, ILP wall*) – trudności wykrycia dostatecznej równoległości w pojedynczym strumieniu instrukcji, zapewniającej pełne wykorzystanie mocy procesora,
 3. "mur" mocy (ang. *power wall*) – wzrost zużycia mocy zależy wykładniczo od częstotliwości (*consuming exponentially increasing power with each factorial increase of operating frequency*)

Użycie wielu rdzeni pozwala na zwiększenie wydajności, ale konkurencyjnym trendem jest dalsza integracja funkcji we/wy w pojedynczym chipie.

Procesor wielordzeniowy – architektura

Procesory jednorodne mają identyczne rdzenie, rdzenie procesora niejednorodnego:

- a) obsługują różne zbiory instrukcji (np. *AMD Accelerated Processing Unit*)
- b) obsługują te same zbiory instrukcji z różną wydajnością i poborem energii (np. *ARM big.little*).

Rdzenie mogą mieć różne architektury (VLIW, superskalarna, wektorowa).

Zyski wydajności– prawo Amdahla, heurystyczna reguła Pollack’a (przyrost wydajności jest proporcjonalny do pierwiastka z liczby rdzeni).

Współczynnik przyśpieszenia może być większy od liczby rdzeni, jeśli uda się zintensyfikować użycie cache L1 (zwarte programy i struktury danych).

Powiązanie rdzeni: mocne (ang. *tightly coupled*) lub słabe (ang. *loosely coupled*).

Rdzenie mogą współdzielić bufor cache

Komunikacja międzyrdzeniowa: przesyłanie wiadomości (ang. *message passing*) lub pamięć współdzielona (ang. *shared memory*).

Powszechnie używane typy połączeń rdzeni: magistrala, pierścień, dwuwymiarowa siatka (ang. *mesh*), macierz połączeń bezpośrednich (ang. *crossbar*).

Multicore i manycore

Rdzenie procesora **multicore** – procesory o dużej wydajności (architektura VLIW lub superskalarna, głębszy potok, większe bufony cache, niekolejne wykonanie)

Liczba rdzeni jest niewielka (2,4,8), działanie często wspomagane przez akcelerator manycore (GPU)

Procesory **manycore** zawierają **wiele prostych niezależnych rdzeni** (10, 100, 1000,...). w celu uzyskania wysokiego stopnia przetwarzania równoległego. Powszechnie używane w systemach wbudowanych i przetwarzaniu macierzowym. Np. chiński Sunway TaihuLight zawiera 40,960 260-rdzeniowych SW26010.

Manycore są zoptymalizowane pod względem równoległości i przepustowości oraz małego zużycia energii kosztem niższej przepustowości pojedynczych wątków

Wymaganie spójności *cache* ogranicza skalowanie procesorów wielordzeniowych.

W systemach *manycore* problem jest omijany dzięki komunikacji “*message passing*” lokalnym pamięciom (NUMA) lub tylko odczytywalnych pamięciach podręcznych (ROM cache) lub fizyczne rozdzielenie partycji.

Podstawowe kroki w tworzeniu aplikacji równoległych

Dekompozycja (ang. *partitioning*)

Podział na dużą liczbę małych zadań wykonalnych współbieżnie (ang. a *fine-grained decomposition of a problem*).

Komunikacja

Zadania najczęściej nie mogą być wykonywane niezależnie z uwagi na powiązania danych, które muszą być transferowane

Grupowanie (ang. *agglomeration*)

Weryfikacja podziału na zadania w kontekście konkretnej architektury – mniej większych zadań to mniej intensywna komunikacja

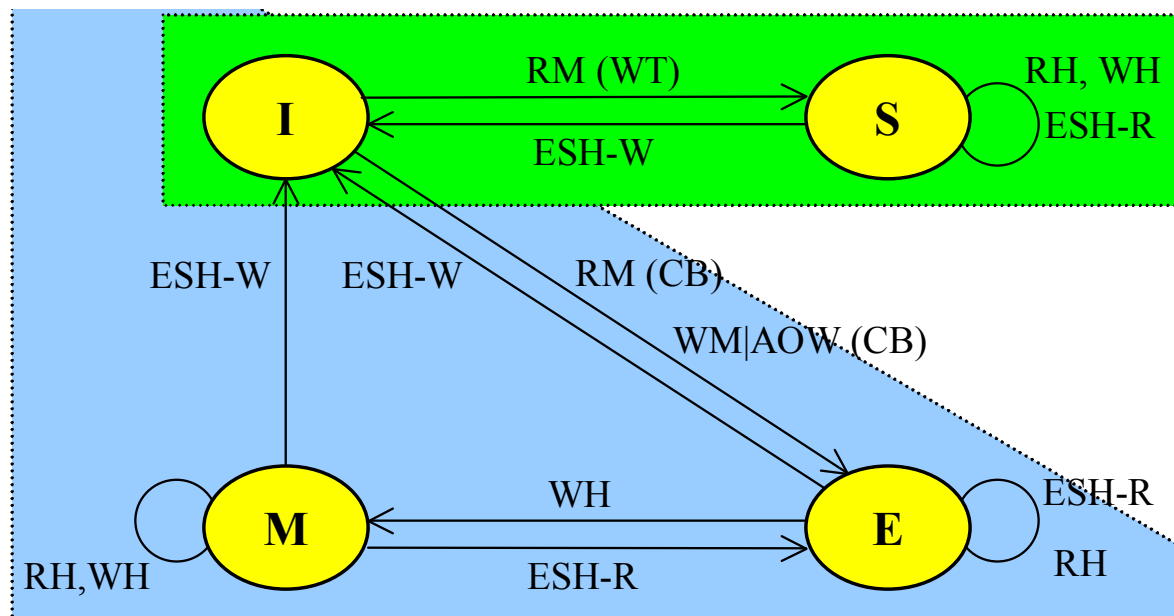
Odwzorowanie (ang. *mapping*)

Przypisanie zadań do jednostek wykonawczych (w systemach z pamięcią współdzieloną szeregowanie zadań (ang. *task scheduling*) jest automatyczne).

W systemach wbudowanych zadania są z natury rzeczy specjalizowane i ich dystrybucja w systemie wieloprocessorowym jest naturalna, a oprogramowanie zadań jest dostosowane do sprzętu.

Podobne efekty występują w systemach cyfrowego przetwarzania sygnałów.

Model spójności pamięci podręcznej – protokół MESI

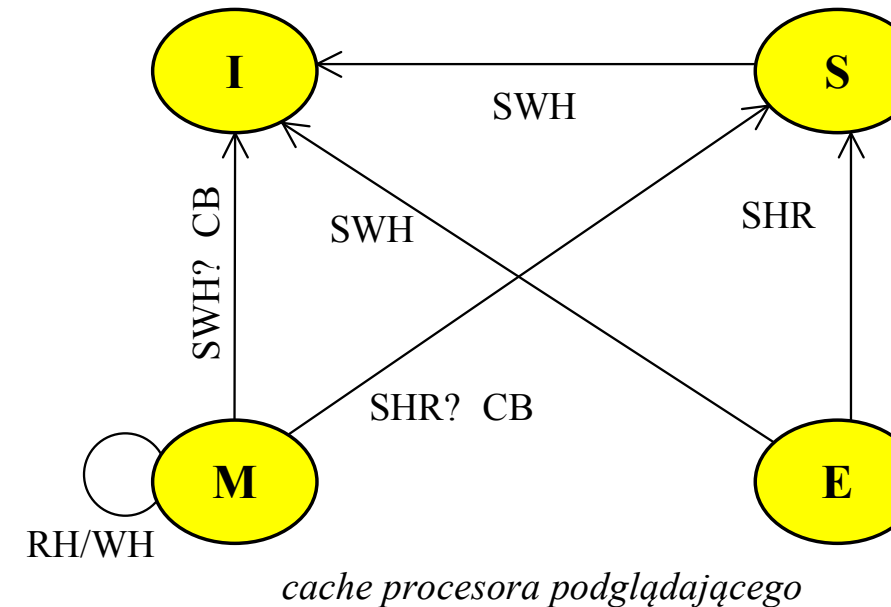
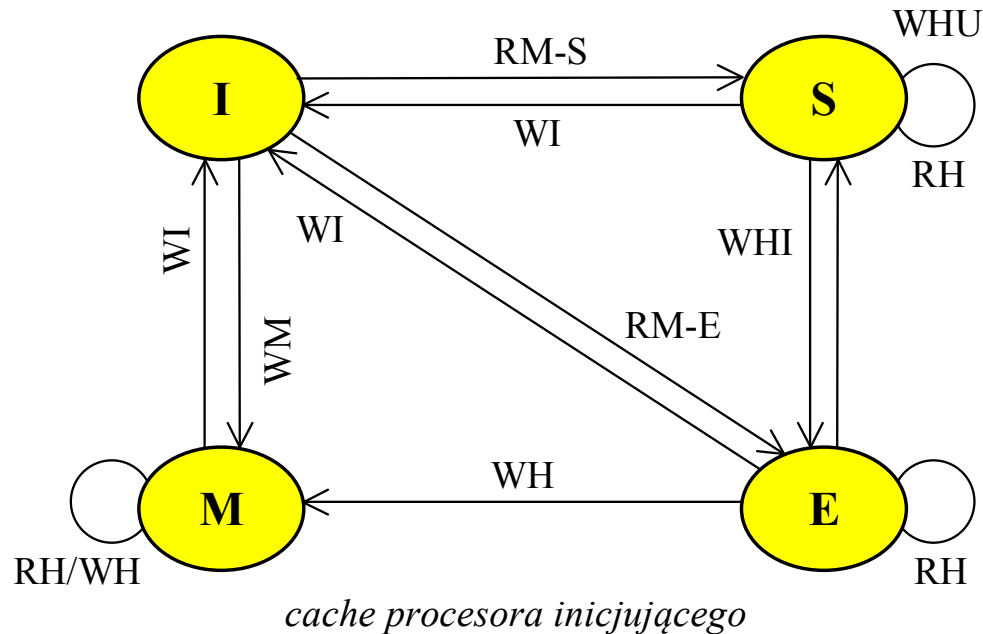


Stany linii pamięci podręcznej obsługiwanej w trybie WT lub CB

(I – nieważny (ang. *invalid*), E – zgodny (ang. *exclusive*), M – zmieniony (ang. *modified*),
S – współdzielony (ang. *shared*))

M – chybiecie (ang. *miss*), H – trafienie (ang. *hit*), R – podczas odczytu, W – podczas zapisu,
ESH – podglądnięcie trafienia zewnętrznego (ang. *external snoop hit*)

Protokół MESI w systemach wieloprocessorowych



Model spójności wielopoziomowej pamięci podręcznej (protokół MESI)

Protokół z podglądem (ang. *snoopy protocol*): sterownik cache musi rozpoznawać kiedy linia jest współdzielona z innymi buforami cache. Aktualizacja linii współdzielonej musi być rozgłoszona innym sterownikom.

protokół „*write update*” – wszystkie współdzielone są aktualizowane

protokół „*write invalidate*” – wszystkie współdzielone są unieważniane