

UNIwersytet w Białymstoku  
Wydział Matematyki i Informatyki  
Instytut Informatyki

Damian Stępkowski

HISTORIA I KATALOG PROCESORÓW  
WIELORDZENIOWYCH W PHP  
I MySQL

*Praca dyplomowa napisana  
pod kierunkiem  
dr Mariusza Żynela*

Białystok 2018

# Spis treści

Spis rysunków	iii
Wstęp	1
<b>1 Pojęcie, rozwój, wady i zalety, budowa procesora wielordzeniowego</b>	<b>3</b>
1.1 Pojęcie procesora wielordzeniowego . . . . .	3
1.2 Historia procesorów wielordzeniowych . . . . .	4
1.3 Rozwój procesora wielordzeniowego . . . . .	5
1.4 Zalety i wady procesora wielordzeniowego . . . . .	6
1.5 Budowa i gniazdo procesora wielordzeniowego . . . . .	8
<b>2 Procesory kaflowe, procesory heterogeniczne i procesory kompozytowe</b>	<b>10</b>
2.1 Architektura procesora kaflowego . . . . .	10
2.2 Architektura procesora wielordzeniowego Raw . . . . .	12
2.3 Procesory wielordzeniowe heterogeniczne . . . . .	13
2.4 Procesory wielordzeniowe kompaktowe . . . . .	17
<b>3 Architektura i implementacja wielowątkowa, system operacyjny, pamięć podręczna</b>	<b>19</b>
3.1 Architektura wielowątkowa . . . . .	19
3.2 Implementacja architektury wielowątkowej . . . . .	21
3.3 System operacyjny . . . . .	24
3.4 Pamięć podręczna . . . . .	25
<b>4 Instrukcja korzystania z Katalogu Procesorów Wielordzeniowych</b>	<b>27</b>
4.1 Ogólny opis funkcjonalności strony głównej i baza danych . . .	27
4.2 Ogólny opis funkcjonalności dla Gościa . . . . .	29
4.3 Ogólny opis funkcjonalności dla Użytkownika . . . . .	32
4.4 Ogólny opis funkcjonalności dla Administratora . . . . .	34
Słowniczek skrótów	37

**Podsumowanie**

**39**

**Bibliografia**

**40**

# Spis rysunków

1.1	Procesor Intel Core i7. [7]	8
1.2	Gniazdo procesora typu Socket A. [2]	9
2.1	Procesor wielordzeniowy kaflowy. [1]	11
2.2	Procesor wielordzeniowy RAW. [1]	12
2.3	Schemat blokowy płytek heterogenicznych. [1]	14
2.4	Łącza dla czterech sieci. [1]	16
2.5	Plan piętra wysokiego poziomu z trzema możliwymi konfiguracjami CLP. [1]	18
3.1	Model wielowątkowej architektury. [1]	21
3.2	Implementację wielowątkowej architektury. [1]	22
3.3	Nierównowaga obciążenia procesora. [1]	23
3.4	Hierarchia pamięci podręcznej. [1]	26
4.1	Strona główna.	27
4.2	Dalsza część strony głównej.	28
4.3	Baza danych.	28
4.4	Menu dla Gościa.	29
4.5	Rozwój procesora.	29
4.6	Pojęcie procesora.	30
4.7	Historia.	30
4.8	Zalety i Wady.	31
4.9	Budowa Procesora.	31
4.10	Rejestracja.	32
4.11	Logowanie.	32
4.12	Panel Użytkownika.	33
4.13	Katalog procesorów wielordzeniowych.	33
4.14	Porównanie.	34
4.15	Logowanie Admina.	34
4.16	Panel Admina.	35
4.17	Edycja procesora.	35
4.18	Dodawanie procesora.	36

# Wstęp

Głównym celem mojej pracy licencjackiej było stworzenie strony internetowej, na której jest przedstawiony katalog do porównywania procesorów wielordzeniowych. Aby porównać procesory trzeba się zarejestrować, po rejestracji można przeglądać procesory i porównać ze sobą wybierając dwa lub trzy procesory. Na stronie również są umieszczone podstawowe informacje o procesorach wielordzeniowych, które można przeczytać bez konieczności zakładania konta.

Do stworzenia strony internetowej został wykorzystany:

- „Sublime Text”- jest to wieloplatformowy edytor tekstu do programowania aplikacji.
- HTML – jest to język do tworzenia stron internetowych, który umożliwia opisywanie struktury strony.
- CSS – jest językiem stylów, który jest używany do wyglądu strony internetowej i prezentuje dokument HTML
- MySQL – system, który zarządza relacyjną bazą danych
- PHP – jest to internetowy język programowy, za pomocą niego możemy tworzyć skrypty po stronie serwera WWW.
- JavaScript – używa się do polepszenia interaktywności, walidacji danych, które zostały wprowadzone do formularza.

Praca licencjacka jest podzielona na cztery rozdziały. W rozdziale pierwszym wyjaśnione jest pojęcie procesora wielordzeniowego, który jest pojedynczym elementem obliczeniowym z dwoma lub więcej samodzielnymi rdzeniami. Tutaj również jest opisana historia procesorów, przedstawione jest powstanie pierwszego procesora i jak sobie radzono gdy nie było procesorów wielordzeniowych. Zarówno opisany jest rozwój procesorów i jak rozwija się architektura wielordzeniowa. Także omówione są zalety, wady i budowa procesora.

Natomiast w drugim rozdziale przedstawiona jest architektura kaflowa procesorów wielordzeniowych. Kaflowe procesory posiadają kilka małych rdzeni zamiast jednego dużego. Została również omówiona architektura prototypu procesora wielordzeniowego Raw. Przedstawione w tym rozdziale są też procesory heterogeniczne i kompaktowe.

W trzecim rozdziale opisana jest architektura wielowątkowa i implementacja architektury wielowątkowej. Przedstawiony jest system operacyjny, który korzysta z wielozadaniowości. Tutaj także została omówiona hierarchia pamięci podręcznej.

Ostatni rozdział, czyli czwarty, jest poświęcony na przedstawienie części praktycznej. Rozdział ten stanowi podręcznik użytkownika utworzony za pomocą screenów i opisów. Tutaj również jest opisana baza danych.

# Rozdział 1

## Pojęcie, rozwój, wady i zalety, budowa procesora wielordzeniowego

### 1.1 Pojęcie procesora wielordzeniowego

Procesor wielordzeniowy jest pojedynczym elementem obliczeniowym z dwoma lub więcej samodzielnymi rdzeniami [3]. Rdzeń jest fizyczną częścią procesora, która jest odpowiedzialna za wykonanie operacji obliczeniowych. Jedną cechą podstawową procesora jest określona długość (liczba bitów) słowa, na którym wykonuje podstawowe operacje obliczeniowe. Wydajność i szybkość procesora wielordzeniowego podnosi się poprzez zwiększenie liczby rdzeni. Każdy rdzeń procesora obsługuje co najmniej jedną potokowość instrukcji. Może również obsługiwać więcej instrukcji, wtedy mówi się, że procesor taki posiada  $x$  rdzeni wirtualnych (wirtualne rdzenie są potęgą dwójki). Potokowość jest formą budowy procesorów, która opiera się na podziale logiki procesora. Jest ona odpowiedzialna za proces wykonywania programu czyli rozkazów procesora. Procesory wielordzeniowe są tak zaprojektowane, że nie używane rdzenie mogą być wyłączone.

Wielordzeniowy procesor implementuje proces wieloprotocowy w jednym pakiecie fizycznym. Projektanci mogą łączyć rdzenie w procesorze ciasno lub luźno. Na przykład rdzenie mogą lub nie muszą współdzielić pamięć podręczną i mogą implementować metody komunikacji między rdzeniami oparte na przekazywaniu pamięci współużytkowanej lub komunikatów. Powszechne topologie sieci do połączenia rdzeni zawierają pierścień, magistralę, poprzeczkę i dwuwymiarową siatkę. Jednorodne, wielordzeniowe systemy zawierają takie same rdzenie. Heterogeniczne, wielordzeniowe systemy mają rdzenie które nie są identyczne (np. big.LITTLE ma heterogeniczne rdzenie, które stosują ten sam zestaw instrukcji, podczas gdy AMD Accelerated Processing Units mają rdzenie, które nie stosują tego samego zestawu instrukcji). Rdzenie podob-

nie jak w jednoprocessorowych systemach i systemach wielordzeniowych mogą implementować architektury jak na przykład wektorowe, wielowątkowe, superskalarne lub VLIW (Very long instruction word czyli bardzo długie słowo instrukcji).

Procesory wielordzeniowe są szeroko stosowane w wielu rodzajach aplikacji, w tym w zastosowaniach sieciowych, ogólnych, wbudowanych, grafice (GPU) i przetwarzaniu sygnału cyfrowego (DSP).

Polepszenie wydajności uzyskanej dzięki zastosowaniu procesora wielordzeniowego zależy w dużym stopniu od zastosowanych algorytmów programowych i ich implementacji. Możliwe zyski są limitowane przez część oprogramowania, które może działać równocześnie na wielu rdzeniach. Ten efekt jest opisany przez prawo Amdahla. Prawo Amdahla jest formułą która daje teoretyczne przyspieszenie wykonania zadania przy stałym obciążeniu, którego można oczekiwać od systemu i którego zasoby są udoskonalone. Prawo Amdahla jest często używane w obliczeniach równoległych do przewidywania teoretycznego przyspieszenia przy użyciu wielu procesorów. W najlepszym przypadku, problemy równoległe mogą powodować przyspieszenia w pobliżu liczby rdzeni, aby móc zmieścić się w pamięci podręcznej każdego rdzenia, unikając używania znacznie wolniejszej pamięci systemu głównego. W przypadku większej części aplikacji nie uzyskamy jednak wzrostu wydajności, chyba że programiści włożą odpowiednio dużo wysiłku w ponowne przeanalizowanie całego problemu i dostosują aplikację do wykonywania wielu wątków jednocześnie[3].

## 1.2 Historia procesorów wielordzeniowych

Dawniej, gdy nie było procesorów wielordzeniowych firmy takie jak Intel i AMD próbowały rozwiązać problem wydajności budując komputery z wieloma procesorami[?]. Jednak by zbudować taki komputer, była potrzebna płyta główna, która posiadała wbudowane więcej niż jedno gniazdo procesora. Było to droższe rozwiązanie, ze względu na fizyczny sprzęt potrzebny do stworzenia dodatkowego gniazda procesora w płycie głównej. Stanowiło to większe opóźnienie ze względu na komunikację, która była pomiędzy procesorami. Płyta główna musiała podzielić dane między procesorami, które znajdowały się w oddzielnych miejscach w komputerze, zamiast wysłać wszystko do jednego procesora. Ta odległość fizyczna sprawiała, że czas wykonania programu był dłuższy. Osadzenie procesora na jednym chipie z kilkoma rdzeniami oznacza mniejszy dystans który jest do pokonania, również mogą dzielić ze sobą rdzenie by móc wykonać ciężkie zadania. Między innymi Intel Pentium II i Pentium III zostały zaimplementowane z dwoma procesorami na jednej płycie głównej [5].

Jednym z pierwszych wyprodukowanych procesorów wielordzeniowych w roku 2001 był Power4 IBM. Wykonany był z dwoma rdzeniami na jednej macierzy. Power4 miał prędkość zegara 1,1 i 1,3 GHz. Została również stworzo-



na wersja Power4+, która miała prędkość zegara 1,9 GHz. Procesor ten był używany w komputerach RS/6000 i AS/400. Od tego czasu procesory wielordzeniowe stały się normą i rozwiązaniem wydajności procesorów. Również dodanie większej liczby wątków maskuje długie opóźnianie operacji. Z biegiem czasu procesory wielordzeniowe z dwóch rdzeni przerodziły się do układów tri-3, tetra-4, hexa-6 i okta-8 [4].

W miarę upływu czasu trzeba było stworzyć mocniejsze procesory, dlatego producenci komputerów wpadli na ideę hiper-wątkowania. Za tą koncepcję jest odpowiedzialny Intel. Została ona zrealizowana w 2002 roku na firmowych procesorach serwerowych Xeon, a potem na procesorach Pentium 4. Hiper-wątkowanie jest nadal używane i stanowi główną różnicę pomiędzy układem Intel i5 a układem i7.

### 1.3 Rozwój procesora wielordzeniowego

Podczas kiedy technologia produkcji się rozwija, zmniejszając rozmiar pojedynczych bramek, ograniczenia fizyczne mikroelektroniki opartej na półprzewodnikach stały się jednym z głównych problemów w zakresie projektowym. Fizyczne ograniczenia mogą powodować znaczne problemy z utratą ciepła i ze synchronizacją danych. Aby poprawić wydajność procesora są używane różne metody. Poszczególne metody paralelizmu na poziomie instrukcji (ILP - Instruction Level Parallelizm) jak na przykład superskalarne pipeliningi, są prawidłowe dla wielu aplikacji ale są nieefektywne dla innych aplikacji, które zawierają trudny do przewidzenia kod. Wiele aplikacji są dostosowane do metod równoległego poziomu wątku, a wiele samodzielnych procesorów jest często stosowanych do zwiększenia ogólnego TLP systemu. Kombinacja zwiększonej dostępnej przestrzeni (ze względu na udoskonalone procesy produkcyjne) i popyt na zwiększenie TLP doprowadziło do rozwoju procesorów wielordzeniowych.

Aspekty biznesowe napędzają szybki rozwój architektur wielordzeniowych. Przez dziesięciolecia, możliwy był wzrost wydajności procesora przy ograniczeniu obszaru układu scalonego, co spowodowało zmniejszenie kosztów urządzenia na układzie scalonym. Alternatywnie, w tym samym obszarze możliwe jest zastosowanie więcej tranzystorów w procesorze. Poprawi to funkcjonalność, zwłaszcza w przypadku architektury złożonych zestawów instrukcji (CISC – Complex Instruction Set Computer). Częstotliwość zegara także wzrosła o rząd wielkości w końcu XX wieku - od niewielu megaherców w latach 80 do kilku gigaherców na początku 2000 roku. Jednocześnie ze spowolnieniem tempa poprawy prędkości zegara, zwiększono wykorzystanie przetwarzania równoległego w postaci wielordzeniowych procesorów, żeby móc poprawić ogólną wydajność systemu. Na tym samym mikroukładzie procesora zostało wykorzystanych wiele rdzeni, zapewne dzięki temu wzrosła sprzedaż procesorów dwu lub więcej rdzeniowych. Między innymi Intel opracował 48-rdzeniowy

procesor do badań w chmurze obliczeniowej. Każdy rdzeń ma architekturę x86. Jest to rodzina zgodnych wstecz architektur zestawów instrukcji opartych na procesorze Intel 8086 i jego wariancie Intel 8088. Producenci komputerów od dłuższego czasu wdrażają projekty symetrycznego przetwarzania wieloprocessorowego z wykorzystaniem dyskretnych procesorów. Problemy, które są związane z architekturą procesorów wielordzeniowych i jej obsługa za pomocą oprogramowania są znane bardzo dobrze. Do tego:

- Korzystanie ze sprawdzonej architektury procesora znacznie zmniejsza ryzyko projektowe.
- W kwestii procesorów ogólnego przeznaczenia dużą motywacją dla wielordzeniowych procesorów jest znacznie zmniejszenie wydajności procesora wywołane zwiększeniem skalowalności częstotliwości. Skutkiem są trzy decydujące czynniki:
  1. Bariera pamięci – pogłębiająca się różnica między prędkościami procesora i pamięcią. W efekcie wywołuje to zwiększenie rozmiarów pamięci podręcznej, aby ukryć opóźnienie pamięci głównej. Ostatecznie, zabieg ten pomaga o tyle, iż przepustowość pamięci nie jest wąskim gardłem w wydajności.
  2. Bariera ILP – wzrastająca trudność w znalezieniu wystarczającej równoległości w jednym strumieniu instrukcji, żeby utrzymać wydajny jednorodzeniowy procesor
  3. Bariera energetyczna – trend zużywania wykładniczo rosnącej mocy (także wzrastające ciepło) za każdym razem przy zwiększeniu wydajności roboczej przez czynnik. Bariera energetyczna stanowi problemy z produkcją, wdrażaniem systemu i projektowaniem, które nie zostały zmniejszone w obliczu natężenia wydajności wywołane barierą ILP i barierą pamięci.

Aby kontynuować wzrost wydajności procesorów ogólnego przeznaczenia, producenci tacy jak Intel i AMD skierowali się w stronę procesorów wielordzeniowych. Intel i AMD poświęcają koszty produkcji procesorów wielordzeniowych, żeby uzyskać wysoką wydajność w poszczególnych systemach i aplikacjach. Architektury wielordzeniowe rozwijają się ale rozwijają się również rozwiązania alternatywne. Bardzo silnym zawodnikiem na rynku jest dalsza integracja funkcji peryferyjnych z chipem [3].

## 1.4 Zalety i wady procesora wielordzeniowego

### Zalety:

Bliskość wielu rdzeni procesora zezwala na pracę układu pamięci podręcznej przy znacznie wyższych częstotliwościach zegara niż było by to możliwe, gdyby sygnały przemieszczałyby się poza mikroukładem. Łączenie równoważnych procesorów na jednej matrycy znacząco poprawia wydajność operacji w pamięci podręcznej (alternatywnie: podsłuchiwanie magistrali). To znaczy, że między procesorami sygnały przemieszczają się na nieduże odległości, a zatem nie ulegają degradacji. Sygnały wyższej jakości zezwalają na przesłanie większej ilości danych w danym okresie czasu, dlatego iż pojedyncze sygnały mogą być krótsze i nie wymagają częstego powtarzania.

Projekty procesorów wielordzeniowych wymagają niewielkiej powierzchni płytki drukowanej, znacznie mniej niż wieloprocessorowe projekty SMP. Również dwurdzeniowy procesor zużywa mniej energii niż dwa połączone procesory jednorodzeniowe, głównie ze względu na zmniejszoną moc wymaganą do przesyłania sygnałów zewnętrznych do układu. Ponadto, rdzenie mają wspólne obwody, takie jak interfejs do magistrali (FSB - Front Side Bus) i pamięć podręczna L2 (prywatna lub w pewnym sensie współdzielona). Ze względu na konkurencyjne technologie dostępnych matryc krzemowych, wielordzeniowe procesory mogą wykorzystywać sprawdzone rdzenie procesora.

Wielordzeniowe chipy również osiągają wyższą wydajność przy mniejszym zużyciu energii. Zapewne może to być duży argument w mobilnych urządzeniach, które są zasilane baterią. Dlatego, że rdzenie w procesorze wielordzeniowym są bardziej energooszczędne, układ staje się w dużym stopniu bardziej wydajny od układu mającego pojedynczy potężny monolityczny rdzeń [3].

### **Wady:**

Maksymalizacja całkowitego użycia zasobów obliczeniowych zapewnianych przez wielordzeniowe procesory wymaga dostosowania także systemu operacyjnego (OS), jak również istniejącego oprogramowania aplikacyjnego. Aby aplikacja była w stanie wykorzystać w pełni możliwości procesorów wielordzeniowych musi być odpowiednio napisana z wykorzystaniem wielowątkowości.

Integracja chipa wielordzeniowego może obniżyć wydajność produkcji chipów. Są one również jeszcze trudniejsze w zarządzaniu termicznym niż procesory jednorodzeniowe o mniejszej gęstości. Intel po części rozwiązał ten pierwszy problem, budując czterordzeniowe procesory, łącząc dwa dwurdzeniowe procesory na jednej matrycy ze zunifikowaną pamięcią podręczną. Dzięki temu możemy użyć obojętnie jakich dwóch pracujących matryc dwurdzeniowych, w przeciwieństwie do produkcji czterech rdzeni na jednej matrycy. Wymagane są wszystkie cztery rdzenie, żeby stworzyć procesor czterordzeniowy. Od architektonicznego punktu widzenia, pojedyncze procesory mogą lepiej wykorzystać powierzchnię krzemu niż procesory wielordzeniowe, dlatego aktywność w rozwoju tej architektury może wiązać się z ryzykiem zestarzenia się. W ostateczności moc przetwarzania nie jest jedynym ograniczeniem wydajności systemu. Przetwarzające dwa rdzenie, które dzielą tą samą szynę

systemową i przepustowość pamięci ograniczają rzeczywistą przewagę wydajności. W raporcie z 2009 r. Dr Jun Ni uwidoczniał, iż jeśli pojedynczy rdzeń jest bliski ograniczeniu przepustowości pamięci, to przejście do procesora dwurdzeniowego może dać poprawę o 30% do 70%. Jeżeli przepustowość pamięci nie stanowi problemu, możemy oczekiwać poprawy o 90%. Prawo Amdahla powoduje, że twierdzenie to budzi wątpliwości. Aplikacja szybciej będzie działać na jednorzeniowym procesorze, gdy czynnikiem ograniczającym byłaby komunikacja pomiędzy procesorami [3].

## 1.5 Budowa i gniazdo procesora wielordzeniowego

Procesor wielordzeniowy jest zbudowany jako scalony układ zamknięty w hermetycznej obudowie. Na ogół wyprowadzenia są złożone ponieważ są bardziej odporne na utlenianie. Sercem procesora jest monokryształ krzemu, do którego tworzy się fotolitografia. Wykonuje się w niej szereg warstw półprzewodnikowych, tworzących w zależności od zastosowania, sieć od kilku tysięcy do kilkuset milionów tranzystorów. Połączenia wykonane są z miedzi i aluminium.

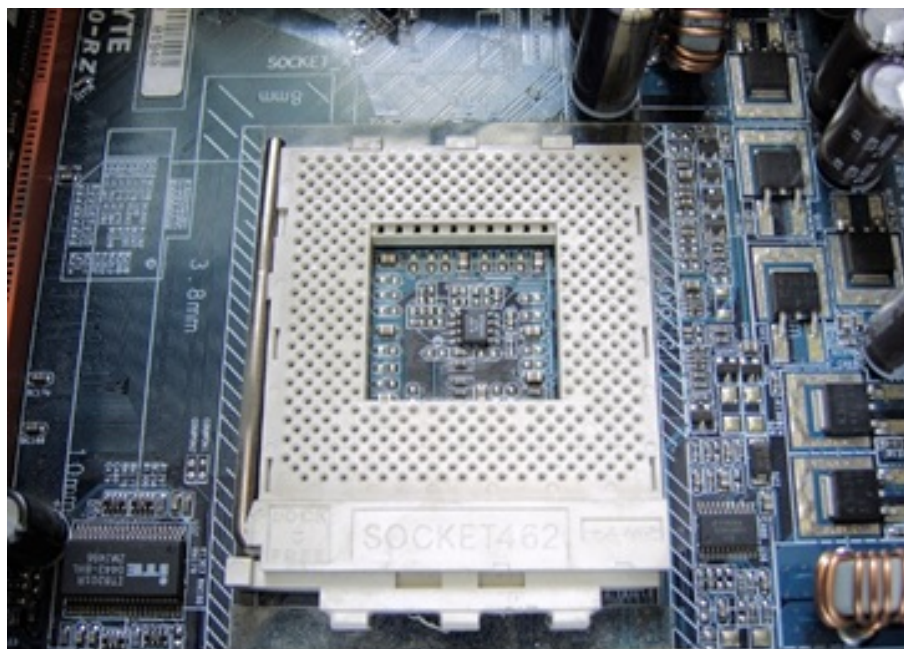


Rysunek 1.1: Procesor Intel Core i7. [7]

Gniazdo procesora jest elementem zapewniającym połączenia mechaniczne i elektryczne pomiędzy urządzeniami a obwodem drukowanym (PCB - Printed Circuit Board). Daje to możliwość na wymianę procesora bez ryzyka uszkodzenia zazwyczaj w trakcie używania narzędzi lutowniczych. W gniazdach procesora wykorzystuje się klipsy przytrzymujące, które są stworzone do stosowania stałej siły, której trzeba użyć gdy urządzenie jest włożone. Gdy w układzie występuje duża liczba pinów, zamiast nich są stosowane gniazda z zerową siłą albo gniazda macierzy sieciowej. Zapewnia to doskonałe rozwiązanie przy uniknięciu zginania szpilek podczas wkładania chipa do gniazda. Gniazda procesora są stosowane w komputerach stacjonarnych i serwerowych. W laptopach zwykle procesory montuje się powierzchniowo, dlatego że łatwiejsza jest wymiana

procesorów.

Gniazdo procesora zbudowane jest z tworzywa sztucznego, metalowej dźwigni lub zatrzasku i metalowych styków dla każdego z pinów. Gniazda są tak zbudowane żeby zapewnić prawidłowe włożenie procesora. Procesory z pakietem PGA są osadzone do gniazda, a zatrzask jest zamknięty [2].



Rysunek 1.2: Gniazdo procesora typu Socket A. [2]

## Rozdział 2

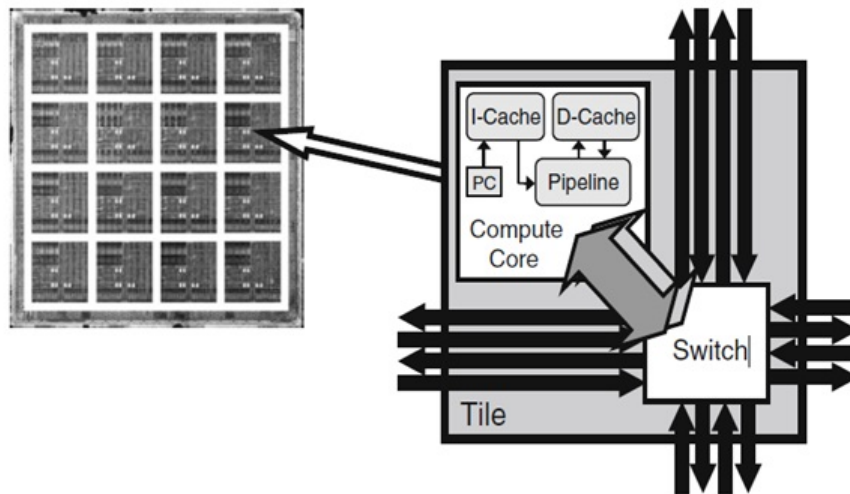
# Procesory kafłowe, procesory heterogoniczne i procesory kompozytowe

### 2.1 Architektura procesora kafłowego

Architektura kafłowa procesorów wielordzeniowych gwarantuje nieograniczoną skalowalność. Dzieje się tak ponieważ prawo Moore'a daje dodatkowe tranzystory na chip. Tak jak wszystkie procesory wielordzeniowe, wielowarstwowe kafele posiadają kilka małych rdzeni obliczeniowych zamiast jednego dużego. Mniejsze rdzenie są wydajniejsze i bardziej energooszczędne niż większe rdzenie. Bardziej funkcjonalne rdzenie mogą być obsługiwane na powierzchni jednego chipa i mogą być wydajniejsze. W przypadku nieobecności innych wąskich gardeł, przepustowość zwiększona jest tak, żeby dopasować liczbę rdzeni dla równoległych obciążeń bez konieczności zwiększenia częstotliwości taktowania. Jednak kluczową koncepcją w kafłach wielordzeniowych jest sposób, w jaki rdzenie procesorów są połączone ze sobą. W kafłowym rdzeniu wielordzeniowym, wszystkie rdzenie są połączone z routerem sieci komunikacyjnej, jak pokazano na rysunku 2.1, tworząc samodzielną modułową "płytkę".

Procesor wielordzeniowy kafłowy składa się z szeregu płytek. Każda taka płytka zawiera niezależny rdzeń obliczeniowy i przełącznik komunikacyjny służący do połączenia rdzeni z płytkami. Procesory kafłowe można łatwo skalować dzięki dodaniu dodatkowych płytek w obszarze układu scalonego i łącząc sąsiednie rutery ze sobą. Dzięki temu tworzy się kompletna sieć komunikacyjna na chipie.

Korzystanie z ogólnego rutera sieciowego z każdej płytki zezwala odróżnić kafele wielowarstwowe od procesorów wielordzeniowych takich jak procesory Intel Core, Cell Broadband Engine i Sun's Niagara. Większość tych procesorów wielordzeniowych posiada rozproszone elementy przetwarzania. Nadal



Rysunek 2.1: Procesor wielordzeniowy kafłowy. [1]

łączą rdzenie ze sobą za pomocą nieskalowanych scentralizowanych struktur, takich jak poprzeczki, interkonekty magistrali i pamięć podręczna. Procesor Cell wykorzystuje sieci pierścieniowe, które są fizycznie skalowane ale również mogą źle funkcjonować z powodu spadku wydajności. Dzieje się tak dlatego, że im większa liczba rdzeni tym większe przeciążenie.

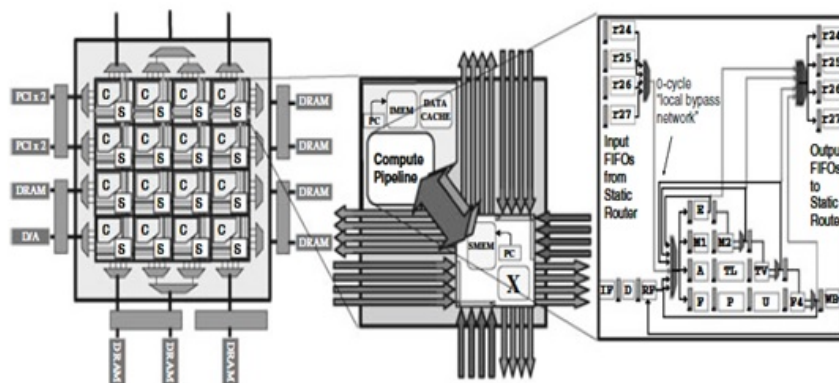
Wielordzeniowe kafle rozprawdają struktury komunikacyjne i obliczeniowe, które zapewniają wydajność, skalowalność i wszechstronność. Mniejsze rdzenie są bardziej wydajne i szybsze ze względu na właściwości skalowania wewnętrznych struktur procesora. Dodatkowo zapewniają tani, szybki dostęp do lokalnych zasobów takich jak podręczna pamięć i ponoszą dodatkowe koszty ale tylko wtedy, gdy są wymagane dodatkowe odległe zasoby. Scentralizowane procesory z drugiej strony zmuszają wszelkie dostępy do ponoszenia kosztów korzystania z jednego ogromnego rdzenia. W dużo mniejszym stopniu dotyczy to także innych wielordzeniowych procesorów ze scentralizowanymi połączeniami. Każdy rdzeń korzysta z pojedynczej dużej magistrali. W kafłowym wieloprocessorze dostęp zewnętrzny jest kierowany przez sieć opartą na chipie i są wykorzystywane tylko segmenty sieciowe między źródłem a miejscem docelowym.

Kafłowe wielordzeniowe architektury są specjalnie zaprojektowane do skalowania, a ze względu na ulepszenie technologii procesorowej zapewniają więcej tranzystorów na każdym chipie. Ponieważ kafle wielordzeniowe wykorzystują rozproszone struktury komunikacyjne i rozproszone obliczenia Procesory różnej wielkości można zbudować kładąc dodatkowe płytki. Przejście do nowszej generacji procesora nie wymaga przeprojektowywania lub ponownej weryfikacji płytki procesora. Oprócz przyszłej skalowalności, ta właściwość ma w dzisiejszych czasach duże zalety w zakresie kosztów projektowania. Aby zaprojektować taki ogromny chip z miliardem tranzystorów wystarczy roz-

mieścić i zweryfikować małą względnie prostą płytkę. Jeśli nie przyniesie to skutków całą operację należy powtórzyć aby wypełnić obszar matrycy. Rdzenie wielordzeniowe ze scentralizowanym połączeniem pozwalają na ponowne wykorzystanie większości procesorów podstawowych. Mimo tego nadal wymagają niestandardowego układu dla każdego rdzenia [1].

## 2.2 Architektura procesora wielordzeniowego Raw

Procesor Raw jest kaflowym prototypem wielordzeniowym. Opracowany w „Computer Architecture Group at MIT” w latach 1997-2002 jest jednym z pierwszych wielordzeniowych procesorów. Architektura procesora Raw obsługuje złącze ISA, które zapewnia równoległy interfejs do bramki i okablowania chipa poprzez odpowiedni wysoki poziom abstrakcji. Jak pokazano na rysunku 2.2 procesor Raw przedstawia liczne zasoby bramek.



Rysunek 2.2: Procesor wielordzeniowy RAW. [1]

Wielordzeniowy procesor Raw składa się z 16 płytek. Każda płytka zawiera procesor komputera, routery, przewody sieciowe oraz pamięć instrukcji i danych. Procesor obliczeniowy wykorzystuje konwencjonalny ośmiopozomowy potok MIPS z interfejsami do sieciowych routerów połączonych bezpośrednio z układem ścieżek obwodowych. Dzielą one użyteczny obszar krzemowy na tablicę 16 jednakowych płytek programowych. Każda taka płytka zawiera routery komunikacyjne i rdzeń przetwarzania, żeby móc połączyć je z sąsiednimi płytkami. Rdzeń przetwarzania zbudowany jest z :

- ośmiopozomowego jednokładowego potoku przetwarzania MIPS (przedstawia to prawa strona rysunku 2.2),
- czteroetapowej jednostopniowej potkowej jednostki FPU,



- podręcznej pamięci danych 32KB,
- instrukcji zarządzania przez 32KB pamięci podręcznej dla potoku przetwarzania.

Stosowane są dwa typy routerów komunikacyjnych (dynamiczny i statyczny). Każda płytką ma taki rozmiar żeby ilość czasu przez który przechodzi sygnał była jednym cyklem zegarowym. Prawdopodobnie przyszłe płytki procesora Raw będą miały setki lub nawet tysiące płytek.

Płytki są połączone czterema 32-bitowymi pełno duplexowymi sieciami scalonymi składającymi się z ponad 12 500 przewodów (rysunek 2.2). Są dwie sieci: statyczne i dynamiczne: statyczne trasy określone podczas kompilacji, a dynamiczne trasy określone podczas wykonywania. Każda płytka jest połączona tylko z jej czterema sąsiadami. Wszystkie przewody są zarejestrowane na wejściu do płytki docelowej. Daje to, że najdłuższy przewód w systemie nie jest dłuższy od długości czy szerokości płytki. Właściwość ta zapewnia wyższe szybkości zegara i ciągłą skalowalność architektury.

Konstrukcja interkonektu Raw na złączu i jego interfejs z potokiem przetwarzania są kluczowymi innowacyjnymi funkcjami. Sieci które opierają się na chipie są narażone na działanie oprogramowania za pośrednictwem Raw ISA. Daje to możliwość kompilatorowi lub programistom możliwość okablowania procesora. Efektywne opóźnienie przewodowe jest nie chronione przed użytkownikiem jako przeskok w sieci. Droga między przeciwległymi rogami procesora zajmuje sześć przeskoków co odpowiada około sześciu cyklom opóźnienia przewodu. Aby pomniejszyć opóźnienie przenoszenia danych skalarnych między płytkami (co stanowi krytyczne znaczenie dla ILP), sieci oparte na procesorze są mapowane nie tylko w rejestrze, ale również bezpośrednio w ścieżkach obejściowych potoku procesora. W rejestrze mapowane porty umożliwiają instrukcję umieszczenia wartości w sieci bez żadnego narzutu. Podobnie, instrukcje używające wartości z sieci są po prostu odczytywane z odpowiedniego rejestru. Programowalne przełączniki odpowiadają za operacje wysyłanie w sieci [1].

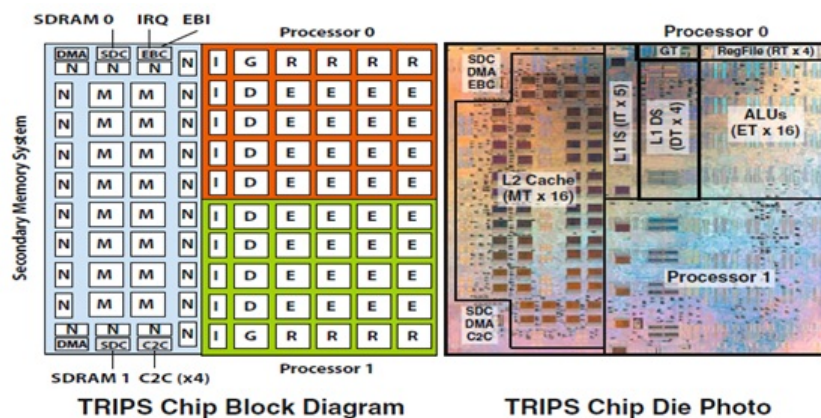
## 2.3 Procesory wielordzeniowe heterogeniczne

Procesory heterogeniczne znane są również jako asymetryczne procesory wielordzeniowe. Architektura heterogeniczna ma na celu poprawić wydajność i efektywność energetyczną. Heterogeniczne procesory skutecznie rozwiązują problem z opcjami rozdrobienia. Procesory rozwiązują ten problem rozdrobienia w czasie projektowania, który będzie wykonywał nieoptymalnie obciążenie pracy w tym samym czasie kompilując zadanie programisty.

Kumar et al. początkowo wystąpił z propozycją architektury heterogenicznej ISA, która stworzona jest z instancji różnych generacji procesorów

CompaqAlpha. Efektywność energetyczną można zmniejszyć, włączając program w najmniej wydajnym rdzeniu wymaganym do spełnienia wymagań wydajności. Ghiasi, Grunwald i Grochowski wykorzystują również architekturę heterogeniczną dla wydajności energetycznej. Kumar et al. rozszerzył również zakres swojej pracy do przestrzeni wieloprogramowej. Ustalił, że asymetria rdzenia zapewnia duże korzyści związane z przepustowością. Annavaram et al. zbadał żywność heterogeniczności w celu przyspieszenia programów wielowątkowych. Heterogeniczność trafia również do strefy komercyjnej. Na przykład w przypadku procesora IBM Cell zawiera jeden średni procesor PowerPC do wykonywania kodu systemowego i ośmiu SPU do wykonywania kodu równoległego do danych. Trend ten prawdopodobnie będzie kontynuowany ponieważ wyspecjalizowane rdzenie graficzne lub kryptograficzne są wbudowane w matrycę procesorów. Chociaż takie wyspecjalizowane procesory są wydajne w przydzielaniu zadań, nie zapewniają adaptacji do różnych ziarnistości równoległości.

W odróżnieniu od jednorodnych płytek architektury wieloprocessorowej takich jak Raw, architektura TRIPS składa się z płytek heterogenicznych. Płytki współpracują tworząc większą jednolitą pamięć podręczną drugiego poziomu i większe procesory. Na rysunku 2.3 przedstawiony jest schemat blokowy na poziomie płytki prototypu TRIPS. Główne trzy elementy układu to dwa procesory i dodatkowy system pamięci, z których każdy jest połączony wewnętrznie przynajmniej z jedną liczbą mikro sieci [1].



Rysunek 2.3: Schemat blokowy płytek heterogenicznych. [1]

Każdy rdzeń procesora jest implementowany przy pomocy pięciu unikalnych płytek:

- jednej globalnej płytki kontrolnej (GT)
- czterech płytek rejestrów (RT),
- pięciu płytek instrukcji (IT)

- czterech płytek danych (DT)
- 16 płytek wykonawczych (ET)

Główna mikroprocesorowa sieć rdzeniowa jest siecią operandów (OPN) rysunek 2.4. Łączy wszystkie płytki z wyjątkiem IT w dwuwymiarowej, prowadzonej w tunelach, topologia siatki 5x5. OPN posiada oddzielne kanały kontrolne i kanały danych i może dostarczać jeden 64-bitowy operand danych na łącze na cykl. Pakiet nagłówka sterowania jest uruchamiany jednym cyklem przed pakietem danych użytecznych w celu przyspieszenia wybudzenia i wyboru pominiętych operandów, które przechodzą przez sieć.

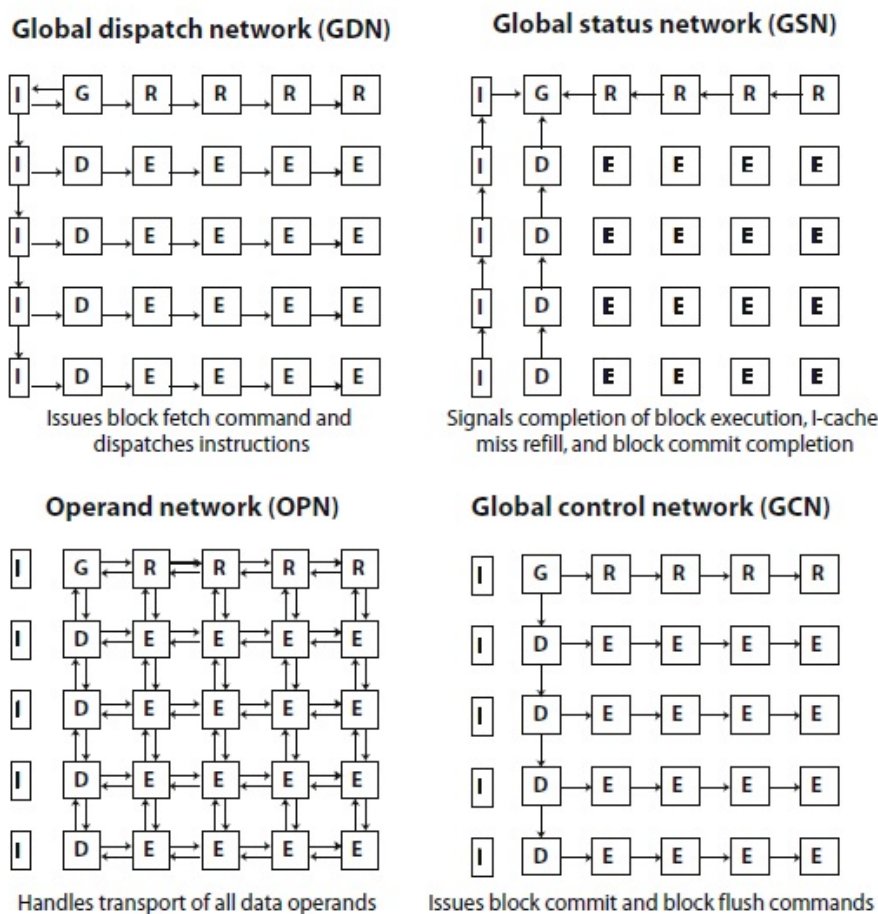
Każdy rdzeń procesora zawiera sześć innych mikro sieci: jeden służy do wysyłania instrukcji czyli globalna sieć wysyłkowa i pięć do kontroli:

- globalna sieć kontrolna (GCN), do zatwierdzenia i opróżnienia bloków
- globalna sieć statusu (GSN), do przesyłania informacji o zakończeniu bloku
- globalna sieć dopełniająca (GRN) do uzupełnienia brakujących wkładów pamięci podręcznej
- sieć stanu danych (DNS) do przekazywania informacji o zakończeniu składowania
- zewnętrzna sieć pamięci (ESN) do określania uzupełnienia pamięci w pamięci podręcznej L2.

Łączy w każdej z tych sieci łączą tylko najbliższe sąsiadujące kafelki, a komunikaty przechodzą przez jeden kafelek na cykl. Rysunek 2.4 przedstawia łącza dla czterech z tych sieci.

Konkretne rozmieszczenie płytek w prototypie tworzy rdzeń z 16-wymiarowym niesprawnym numerem, 64 KB pamięci podręcznej instrukcji L1 i 32 KB pamięci podręcznej danych L1. Zostały rozszerzone pliki rejestru i logika sterowania, aby obsługiwać również cztery wątki SMT. Mikroarchitektura obsługuje jednocześnie do ośmiu bloków TRIPS w locie, zapewniając w ten sposób okno w locie do 1024 instrukcji. Kontrola bloku obsługuje dwa tryby: jeden blok niespekulatywny i siedem bloków spekulatywnych.

Dwa procesory komunikują się za pośrednictwem systemu pamięci wtórnej, w którym osadzona jest sieć na chipie (OCN). OCN jest to sieć o rozmiarze siatki 4 x 10, z 16-bitowymi łączami danych i czterema kanałami wirtualnymi. Sieć ta jest zoptymalizowana pod kątem transferu wielkości pamięci podręcznej. Obsługiwane są również inne rozmiary żądań dla operacji takich jak ładowanie i przechowywanie na stronach nieobsługiwanych. OCN jest budową transportową dla wszystkich pamięci między procesorami, pamięci podręcznej L2, DRAM, we/wy i DMA (bezpośredni dostęp do pamięci)



Rysunek 2.4: Łącza dla czterech sieci. [1]

Fizyczne projektowanie i implementacja układu TRIPS były oparte na zasadach partycjonowania i replikacji. Chip bezpośrednio odpowiada logicznej hierarchii płytek TRIPS, które są połączone tylko punktami z siecią najbliższych sąsiadów. Jedynymi wyjątkami dla komunikacji najbliższego sąsiada są globalne sygnały resetowania i przerwania, które są tolerancyjne na opóźnienia w wielu etapach w całym układzie.

Układ TRIPS jest zaimplementowany w procesorze IBM CU-11 ASIC, który ma narysowany rozmiar operacji wynoszący 130 nm i siedem warstw metalowych. Sam układ zawiera ponad 170 milionów tranzystorów i 1,06 km trasowania kabla, w obszarze chipów 18,3 mm na 18,4 mm umieszczonym w pakiecie z kwadratową siatką o wielkości 47,5 mm. Na rysunku 2.4 pokazano opisowe zdjęcie matrycy układu TRIPS, pokazujące granice płytek TRIPS. Oprócz płytek rdzenia, TRIPS obejmuje również sześć sterowników, które są podłączone do reszty systemu za pośrednictwem sieci on-chip (OCN). Dwa kontrolery DDR SDRAM 133/266 MHz (SDC), z których każdy łączy się z indywidualnym modulem DIMM 1 GB SDRAM. Sterownik do układu scalone-

go (C2C) rozszerza sieć komputerową na czteroportowy router sieciowy, który łączy się z innymi układami TRIPS. Łącza te nominalnie przebiegają w połowie zegara procesora i do 266 MHz. Dwa kontrolery bezpośredniego dostępu do pamięci (DMA) można zaprogramować do przesyłania danych i z dowolnych dwóch regionów fizycznej przestrzeni adresowej, w tym adresów odwzorowanych na inne procesory TRIPS. Wreszcie, zewnętrzny kontroler magistrali (EBC) jest interfejsem dla procesora kontrolnego PowerPC na poziomie płyty. Aby zmniejszyć złożoność projektu, projektanci postanowili na odciążenie znacznej części systemu operacyjnego [1].

## 2.4 Procesory wielordzeniowe kompaktowe

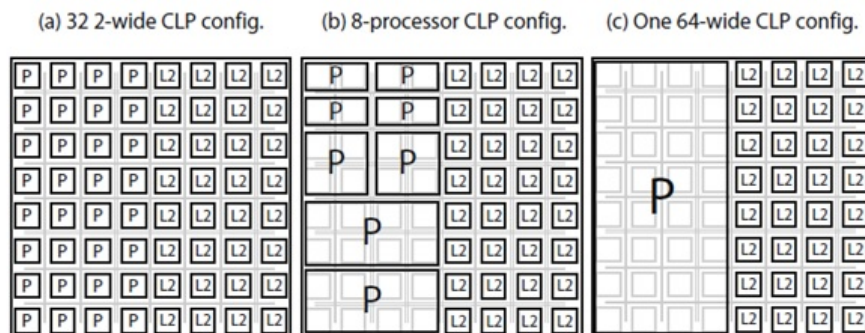
Procesory kompaktowe z elastycznym rdzeniem mają zapewnić adaptację zarówno pod względem liczby i ziarnistości procesorów. Również umożliwiają wydajne wykonywanie dużej i małej liczby zadań. Podstawowym podejściem jest grupowanie wielu mniejszych, identycznych procesorów fizycznych w mniejszym obszarze zagregowanym, w celu utworzenia większych, logicznych procesorów. Komponowanie procesorów logicznych z wielu prostych procesorów daje systemowi możliwość dynamicznego i przejrzystego dostosowywania się do zakresów równoległości na poziomie instrukcji ILP i równoległości wątków TLP do oprogramowania aplikacyjnego. Ponadto procesory kompaktowe, które obsługują skalowanie do wielu rdzeni, mogą zmniejszyć obciążenie programisty dla aplikacji zapewniając wyższą wydajność pojedynczego wątku i redukując liczbę wątków, którymi musi zarządzać programista. Wielordzeniowy procesor złożony z procesorów będzie dobrze działać w aplikacjach z kilkoma wątkami o wysokiej ILP, podczas gdy duża ilość procesorów będzie dobrze działała w aplikacjach z wieloma wątkami o niskiej ILP.

Jednym z głównych przykładów architektury kompaktowej jest Core Fusion. Zapewnia mechanizmy, które pozwalają połączyć ograniczoną liczbę rdzeni - maksymalnie do czterech rdzeni. Główną zaletą Core Fusion jest to, że kompatybilny jest z obecnymi częściami fabrycznymi. Jedną z wad jest scentralizowana funkcja zmiany nazwy rejestru ogranicza jego skalowalność. Jednym z przykładów kompaktowych procesorów jest firma, która łączy wiele rdzeni w kolejności tak by stworzyć procesor, którego bloki konstrukcji są bardziej energooszczędne. Takie procesory osiągają mniejszą złożoną wydajność. Również można przedstawić systemy kompaktowe jako wirtualne hierarchie. Wirtualne hierarchie odpowiadają za elastyczną agregację i izolację poszczególnych rdzeni w celu utworzenia procesorów wielordzeniowych.

Procesor CLP składa się z wielu prostych rdzeni o wąskim zakresie. Rdzenie takie mogą być agregowane dynamicznie by można było utworzyć bardziej wydajny procesor jednowątkowy. W ten sposób liczba i rozmiar procesorów może zostać dostosowywana w locie. Zapewni to osiągnięcie celu, który najlepiej odpowiada potrzebom oprogramowania w danym momencie. Ten sam

wątek oprogramowania może funkcjonować bez modyfikacji binarnych czyli od jednego rdzenia do maksymalnie 32 rdzeni. Oprogramowanie do pracy na niskim poziomie może decydować o tym, jak najlepiej wykorzystać wydajność wątku (TLP) i efektywność energetyczną.

Rysunek 2.5 przedstawia plan piętra wysokiego poziomu z trzema możliwymi konfiguracjami CLP. Małe kwadraty po lewej stronie reprezentują pojedynczy fizyczny rdzeń przetwarzania, natomiast po prawej stronie pokazują pamięć podręczną L2. Jeśli jest możliwa duża liczba wątków, system może uruchomić wtedy 32 wątki, po jednym na każdym rdzeniu (przedstawia to rysunek 2.5 „a”). Jeżeli jest potrzebna wysoka wydajność wątku pojedynczego i wątek ma wystarczającą ILP, CLP może być skonfigurowana tak aby wykorzystywała optymalną liczbę rdzeni i zmniejszy wydajność do 32 (przedstawia to rysunek 2.5 „c”). W celu zmniejszenia poboru energii, na przykład w centrum danych lub w trybie zasilania bateryjnego, system musi skonfigurować procesor CLP. Musi to tak zrobić by uruchamiać każdy wątek w najbardziej energooszczędnym momencie. Rysunek 2.5 „b” przedstawia konfigurację procesora CLP zoptymalizowaną pod względem energii lub mocy z ośmioma wątkami w zakresie procesora [1].



Rysunek 2.5: Plan piętra wysokiego poziomu z trzema możliwymi konfiguracjami CLP. [1]

W pełni kompaktowy procesor nie ma żadnych fizycznych struktur wśród wielu procesorów. Zamiast tego, procesor CLP opiera się na rozproszonych protokołach mikroarchitektonicznych, aby zapewnić niezbędne pobieranie instrukcji, wykonanie, przewidywanie rozgałęzień, przewidywanie zależności i dostęp do pamięci. Pełna kompaktowość jest trudna w konwencjonalnych ISA, ponieważ jednostki atomowe są indywidualnymi instrukcjami. To wymaga, tego aby decyzje kontrolne nie były wykonywane zbyt często, żeby koordynować procesor rozproszony.

## Rozdział 3

# Architektura i implementacja wielowątkowa, system operacyjny, pamięć podręczna

### 3.1 Architektura wielowątkowa

Producenci komputerów poszukują nowych sposobów na efektywne wykorzystanie coraz większej liczby tranzystorów. Konwencjonalny, dynamiczny procesor super skalarny wykorzystuje tranzystory do poprawy wydajności, dzięki wykorzystaniu równoległości na poziomie instrukcji. Procesory wykonują dużo instrukcji równoległe, które nakładają się na jednostki funkcjonalne i opóźniają pamięć. Poprzez równoległe wykonanie kolejnych instrukcji natrafiło na dwa problemy:

1. Procesory komponują pojedyncze okno instrukcji do wyszukiwania samodzielnych instrukcji. Powiększenie okna nie daje gwarancji poprawności wydajności
2. Co więcej wiele kluczowych układów w procesorach skaluje się super liniowo z liczbą instrukcji wykonywanych równoległe z rozmiarem okna.

W przypadku połączenia z szybszymi zegarami, liniowe skalowanie doprowadzi do ogromnej złożoności procesora. Także może doprowadzić to do dużej rozproszonej mocy procesora.

Możemy zapobiec tym dwóm problemom używając wieloprocesorowego układu scalonego CMP. CMP jest skalonym układem mikroprocesorów, który jest zamieszczony na płycie krzemowej. Układy takie wykorzystują wiele rdzeni aby wzrosła liczba tranzystorów. CMP unikają skalowania dzięki użyciu wielu małych rdzeni zamiast jednego dużego rdzenia. Każdy obwód mniejszego rdzenia dobrze radzi sobie z mniejszą liczbą instrukcji. Zapewnia również mniejszy rozmiar okna. Tak więc mniejsze rdzenie zredukują oba problemy

złożoności procesora i rozproszenia mocy. Należy zauważyć, że programowanie równoległe jest trudniejsze od programowania sekwencyjnego. Programy równoległe wymagają zarządzania danymi. Wiąże się to z dystrybucją, przesyłaniem danych oraz synchronizacją. Programowanie równoległe stanowi duże obciążenie dla programisty. Zarządzanie danymi może być trudne i ma duży wpływ na wydajność podczas słabej synchronizacji.

Wieloprocessorowe konwencjonalne rozwiązania są skuteczne mimo potrzeby programowania równoległego. Różnicą między wieloprocessorami a CMP jest to, że wieloprocessory były ukierunkowane na rynki niszowe takie jak serwery baz danych. CMP powinno być programowane przez doświadczonych programistów. Programy sekwencyjne, muszą być automatycznie zrównoleglone przez kompilator. Programowanie sekwencyjne osiągnęło sukces w przypadku aplikacji numerycznych.

Architektury wielowątkowe odnoszą się do zagadnień programowania CMP. Zarówno dotyczy to problemów skalowania jak i ograniczonego paralelizmu procesorów super skalarnych. Spekulacyjne procesory wielowątkowe dzielą sekwencyjny program na ciągle fragmenty programów. Programy takie są wykonywane na wielu rdzeniach równoległe. Aczkolwiek zadania nie dają gwarancji niezależności. Wykonanie przebiega przez spekulację, że zadania są niezależne. Bardziej korzystne jest zastosowanie wielu mniejszych rdzeni niż jednego dużego ze względu na procesory wielordzeniowe które unikają skalowania dynamicznych. Dzieje się tak ponieważ, równoległość pomiędzy dalszymi instrukcjami jest większa. Procesory te unikają problemów z ograniczeniem równoległości super skalarnych.

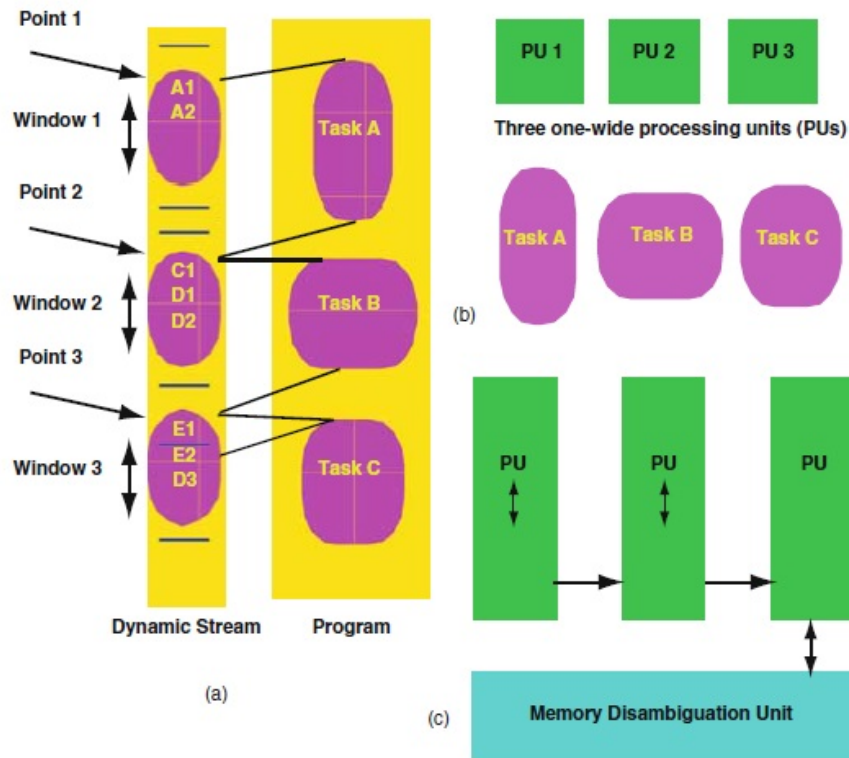
Paralelizm pokazuje, że równoległość występuje w programach sekwencyjnych i nie jest równomiernie rozłożony w dynamicznym strumieniu instrukcji. To znaczy, że instrukcje bliskiej odległości, mogą być zależne. W ten sposób ogranicza się paralelizm w małym oknie instrukcji. Instrukcje które są daleko od siebie, prawdopodobnie będą niezależne. Wielowątkowe architektury zestawiają duże okno instrukcji, łącząc wiele małych podokien. Każde takie podokno jest mapowane i wykonywane na rdzeniu wielowątkowego procesora.

Architektury wielowątkowe wykorzystują kompilator do partycjonowania sekwencyjnego w zadaniach spekulacyjnych. Mimo to takie zadania muszą być niezależne od siebie. System kontroluje przepływ danych i sprawdza czy zadania są niezależne. Zadania są tworzone przez przewidywanie, z jednego zadania na drugie. Zadania są wykonywane równoległe przy założeniu, że nie ma zależności między zadaniami. Każde takie zadanie jest odwzorowane na rdzeń wirtualny lub fizyczny. Rysunek 3.1 przedstawia nam model wykonania prostego przykładu. Zadania w rejestrach i pamięci jest zatwierdzany w kolejności. Same zadania wykonują porządek w programowaniu na całym rdzeniu. Powoduje to dwustopniowe zatwierdzenie hierarchicznie [1].

Model wykonania wielowątkowej architektury przedstawia na rysunku 3.1:

a) Statyczny program podzielony na trzy zadania odpowiadające trzem





Rysunek 3.1: Model wielowątkowej architektury. [1]

oknom.

- b) Zadania przypisane do jednostek rdzenia.
- c) Jednostki przetwarzania połączone są w celu przekazywania wartości między zadaniami. Jednostka ujednoznaczniająca pamięć służy do obsługi zależności pamięciowych.

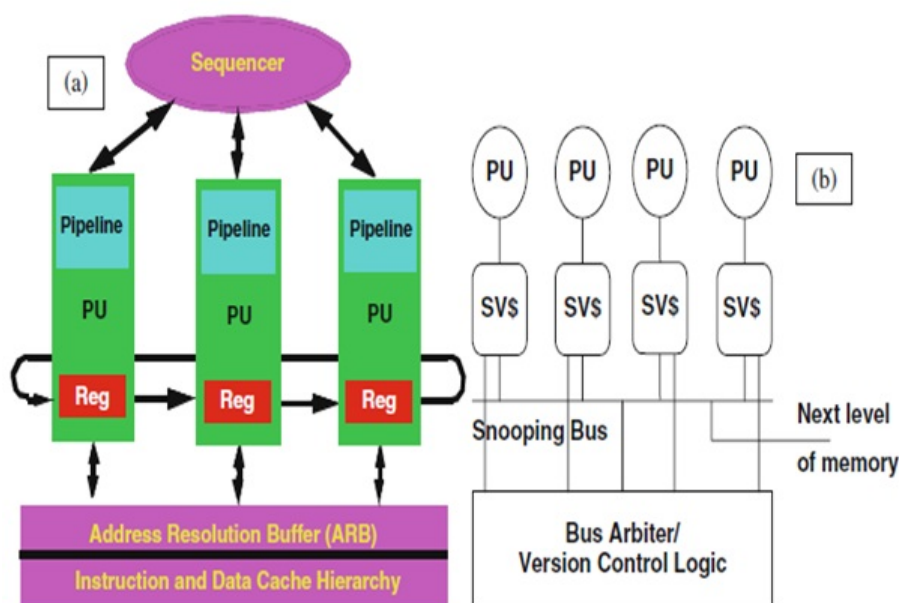
## 3.2 Implementacja architektury wielowątkowej

Każdy z rdzeni posiada osobisty plik rejestru. W zależności od implementacji taki rdzeń może mieć prywatną pamięć podręczną. Również rdzeń może mieć jedną pamięć podręczną współdzieloną przez rdzenie.

W celu zaimplementowania modelu wykonania, odpowiedzialność za przepływ sterowania ponosi wartość rejestrów. Wtedy pamięć jest podzielona pomiędzy sprzętem a kompilatorem. Podczas gdy kompilator jest używany do partycjonowania programu sekwencyjnego w zadaniach, dynamiczna wielowątkowość używa sprzętu do partycjonowania zadań. Zmierza do tego, że odradza nowe zadanie po wywołaniu pętli lub funkcji. W momencie przepływu kontroli między zadaniami każde z takich zadań musi przewidzieć następne do

wykonania zadanie. Zadania mogą potencjalnie prowadzić do więcej niż dwóch następnym zadań. Istnieją dwie możliwości:

- a) Kompilator jawnie zamieszcza listę następców dla każdego zadania, czyli do pewnego maksimum.
- b) Tabela sprzętowa dowiaduje się o następcach z poprzednich zadań tak, że tabelę można ograniczyć do maksimum.



Rysunek 3.2: Implementację wielowątkowej architektury. [1]

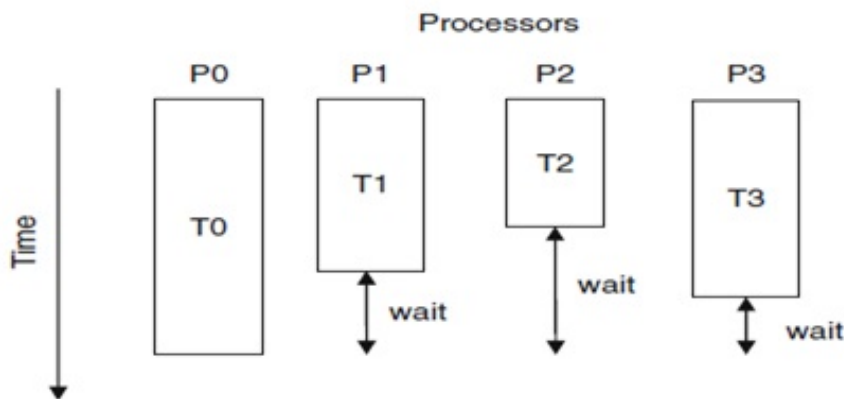
Powyższy obrazek przedstawia implementację wielowątkowej architektury. Część (a) obrazka przedstawia sekwencję zadań, które wykonują przewidywanie między zadaniami. Każda jednostka rdzenia posiada własny plik rejestru i potoku. Dostęp pamięci prowadzi do adresu bufora rozdzielczości (ARB) w celu zjednoczenia pamięci. ARB jest wspierany przez konwencjonalną hierarchię pamięci. Część (b) obrazka przedstawia rozproszoną alternatywę dla scentralizowanego ARB nazywaną spekulacyjną pamięcią podręczną.

Zadania posiadają wspólną logiczną przestrzeń rejestrów. Fizyczne pliki rejestrów są prywatne dla każdego rdzenia. Takie prywatne pliki rejestru dają możliwość na zamianę nazw rejestrów między zadaniami. Możliwość zmiany nazwy w zadaniami odbywa się dzięki standardowemu rejestrowi każdego z rdzeni. Takie zmienianie nazwy przez zadania daje możliwość każdemu plikowi rejestru rdzenia na buforowanie zadania spekulacyjnego. Wartości są używane do aktualizacji stanu architektonicznego.

Ważne komponenty implementacji wynikają z modelu wykonania, organizacji wysokiego poziomu i wyboru zadań. Również wybranie zadań ma znaczenie dla osiągnięcia wydajności w wielowątkowej architekturze. Takie partycjonowanie zadań uwzględnia kluczowe parametry wydajności w procesorach:

- Zależność danych między zadaniami czyli pamięci i rejestru
- Brak równowagi obciążenia
- Zależność między zadaniami
- Przesyłanie obiektu ze względu na duże zadania

Zadania zatwierdzają się w przypadku programu które wykonują znacznie dłuższe zadania i mogą zakończyć wykonywanie przed poprzednim zadaniem. Dzieje się tak, pod warunkiem, że poczekają do momentu kiedy poprzednie zadanie zostanie zatwierdzone. Takie czekanie jest notacją formalną nierównowagi obciążenia, tak jak przedstawiono na rysunku 3.3. Zmniejszenie nierównowagi obciążenia nie wymaga, aby wszystkie zadania były podobne wielkością do siebie. Zadania sąsiadujące nie różnią się znacząco wielkością. Duże zadania mogą prowadzić do przesyłania obiektu. Może być tak, że zabraknie miejsca na buforowanie wartości spekulacyjnych. Trzeba wtedy poczekać aż nie będzie dostępnego miejsca.



Rysunek 3.3: Nierównowaga obciążenia procesora. [1]

Rysunek 3.3 przedstawia nierównowagę obciążenia. Zadania T1, T2 i T3 muszą czekać na zakończenie zadania T0 i zatwierdzenie zanim będzie można zatwierdzić zadania. Również muszą czekać na zwolnienie jednostki przetwarzania dla innych zadań.

### 3.3 System operacyjny

Jądro systemu operacyjnego jest odpowiedzialne za:

- zarządzanie zasobami,
- planowanie,
- tworzenie sieci,
- integrację urządzeń
- zarządzanie wieloma niskopoziomowymi interfejsami sprzętu.

Większość współczesnych systemów operacyjnych intensywnie korzysta z wielozadaniowości. Dodanie wielu rdzeni dodaje nowy wymiar do zakresu na poziomie systemu, którym system operacyjny musi zarządzać. Wewnątrz systemu operacyjnego kilka kluczowych struktur danych ułatwia systemowi śledzenie stanu dynamicznego systemu komputerowego. Struktury danych zarządzają szeregowaniem procesów, pamięcią wirtualną i zarządzaniem wejścia/wyjścia. W symetrycznym systemie wieloprocesorowym system operacyjny może działać na dowolnym procesorze. W niektórych przypadkach systemy operacyjne mogą działać jednocześnie na wielu procesorach. Aby chronić przed szybkością transmisji danych w takich sytuacjach system operacyjny umieszcza konstrukcje blokujące. Taką konstrukcję blokującą umieszcza się w wielu kluczowych wewnętrznych strukturach danych. Wraz ze wzrostem liczby procesorów w systemie wzrasta również potencjał rywalizacji o blokady w kluczowych strukturach. Może to poważnie zaszkodzić ogólnej skalowalności i przepustowości systemu.

W efekcie nowoczesne systemy operacyjne muszą stosować lepsze blokady ziarnistości w strukturach danych. Systemy operacyjne wykazały znaczące korzyści w zakresie skalowalności. Takie korzyści ze skalowalności często wiążą się z obciążeniem procesora. Program systemu operacyjnego jest ciągle ulepszany. Program taki musi rozumieć zarządzanie procesami na wielu procesorach. W przypadku niewielkiej liczby procesorów znacznie szybciej poprawia się reagowanie na poziomie systemu. W sytuacji gdy dostępna liczba procesorów jest bardzo wysoka to wtedy program może stać się wąskim gardłem w aktywnych procesorach.

Wraz z rozwojem systemów oferujących dużo większą moc obliczeniową, wzrasta również zainteresowanie wykorzystaniem technik wirtualizacji. Wirtualizacja w systemie wieloprocesorowym udostępnia logiczne partycjonowanie zasobów pojedynczego systemu fizycznego. A także rdzenie i w wielu wirtualnych systemach, które są izolowane od siebie. Hiperwizor to platforma wirtualizacji, która zezwala na uruchomienie wielu systemów operacyjnych na komputerze w tym samym czasie. Hiperwizor odgrywa kluczową rolę w zarządzaniu niskopoziomowymi zasobami systemowymi. Hiperwizor prezentuje również niezależne obrazy systemu wirtualnego.

W systemach SMP wyższej klasy więcej aplikacji może skorzystać z równoległego przetwarzania. Rozwój i walidacja jakości równoległych aplikacji jest bardzo trudna z uwagi na kilka czynników. Po pierwsze, przeprowadzono już wiele badań nad programowaniem równoległym. Przekształcenie sekwencyjnego programu w równoległy jest trudne. Po drugie, jest trudno udowodnić, że programy są poprawne pod każdym względem. Po trzecie, optymalizacja programów równoległych jest trudna ze względu na ziarnistość blokady oraz synchronizację i komunikację między wątkami. Podstawowe pierwotne operacje wydają się być jednymi z najwolniejszych w nowoczesnych systemach komputerowych.

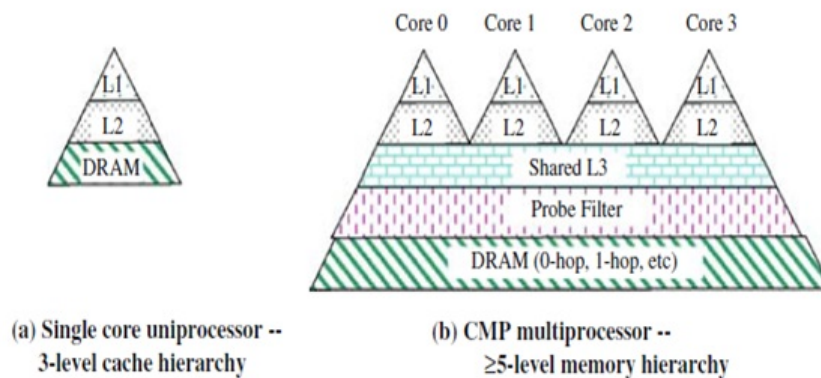
Istnieją również pewne ważne wyzwania związane ze skalowalnością na poziomie systemu z wieloma rdzeniami. Dodanie większej liczby rdzeni zwiększa obciążenie wszystkich współdzielonych zasobów w tym pamięć podręczną, magistralę i porty wejścia/wyjścia. Kluczowe analizy obejmują opóźnienia pamięci, przepustowości pamięci i równoległości pamięci. W przypadku CMP powiększa się zakres i znaczenie szczegółów projektu na poziomie systemowym [1].

### 3.4 Pamięć podręczna

Wraz ze wzrostem liczby rdzeni procesora, istotnych jest kilka ważnych czynników związanych z pamięcią podręczną. Po pierwsze, dla danego rozmiaru matrycy, danego chipa, istnieje zasadniczy kompromis między liczbą rdzeni a rozmiarem pamięci podręcznej. Po drugie, istnieje również kompromis w dystrybucji dostępnego rozmiaru pamięci podręcznej w hierarchii pamięci. Korzystanie z współdzielonej pamięci podręcznej ma największy sens.

W hierarchii pamięci podręcznej występuje pojedynczy rdzeń. Polega to na tym aby rdzeń był jak największy bez dodawania dodatkowych cykli opóźnienia. Również można zmniejszyć częstotliwość taktowania procesora. Wysokie wartości trafień pamięci podręcznej L1 zapewnia zoptymalizowane opóźnienie dla większości dostępu w wielu typowych obciążeniach. Umożliwia to pamięci podręcznej na poziomach hierarchii dokonywać różnych wyborów takich jak pojemność, opóźnienie i przepustowość. Projektowanie CMP w sposób naturalny prowadzi do ponownej oceny tradycyjnego podejścia do zaprojektowania jednoprosesowej hierarchii pamięci podręcznej. Pamięć podręczna wysokiego poziomu jest bardzo duża bez dodawania dodatkowych etapów potoku do rdzenia procesora. Również nie trzeba zmniejszać taktowania częstotliwości procesora.

Po pierwsze, system musi walczyć ze zwiększonym ruchem pamięci, który pochodzi z wielu procesorów. Po drugie, większość procesorów będzie korzystać z prywatnych pamięci podręcznych L1. Obecność wielopoziomowych pamięci podręcznych otwiera możliwość współdzielenia pamięci podręcznej. W przypadku gdy obciążenie równoległe jest jasne, to współużytkowane bu-



Rysunek 3.4: Hierarchia pamięci podręcznej. [1]

forowanie może być pomocne w pobieraniu instrukcji. Jeden wątek skutecznie przesyła kod do współużytkowanej pamięci podręcznej, która jest później używana przez drugi wątek. Gdy wiele wątków czyta wspólną strukturę danych, to współużytkowanie pamięci podręcznej może przynieść korzyści. Współdzielona pamięć podręczna może uniwersalnie reagować na równe wymagania różnych obciążeń. Istnieją pewne wady dzielenia pamięci podręcznej. Uruchomienie wielu obciążeń wymagających dużych rozmiarów pamięci podręcznej może doprowadzić do wymazywania pamięci podręcznej pomiędzy wątkami. Może to powodować nienaturalne zwiększenie aktywności na poziomie systemu. Również obniży ogólną wydajność systemu. Spowoduje to, że aplikacje doświadczą anomalii wydajności i niestabilności systemu. Po trzecie, każdy procesor działa na osobnym wątku wykonywania. Połączony zestaw roboczy systemu jest zwiększony. W systemach obsługujących wirtualizację czyli parę uruchomionych systemów na tej samej platformie, ten efekt jest powiększany przez wiele systemów operacyjnych [1].

# Rozdział 4

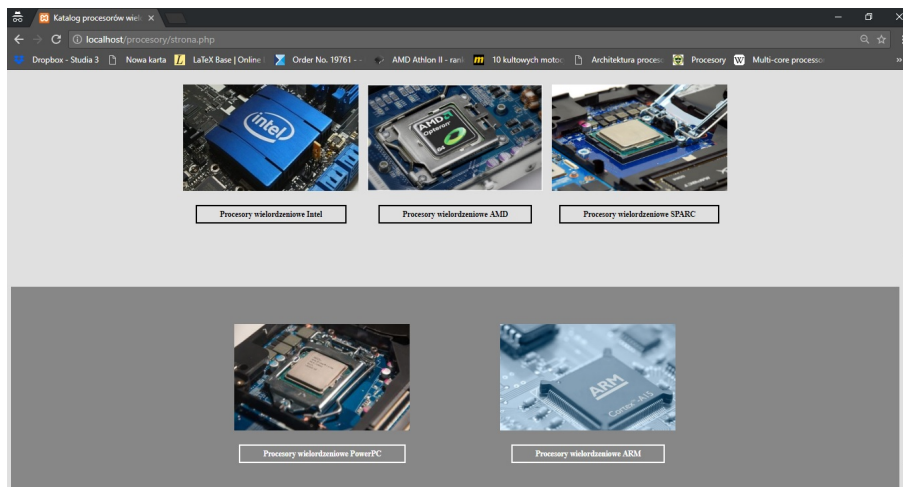
## Instrukcja korzystania z Katalogu Procesorów Wielordzeniowych

### 4.1 Ogólny opis funkcjonalności strony głównej i baza danych



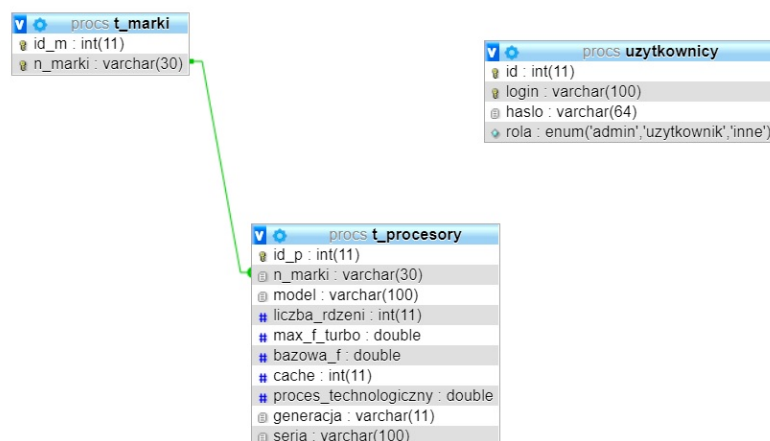
Rysunek 4.1: Strona główna.

Na stronie głównej znajdują się pasek menu. Po wybraniu przycisku możemy przeczytać informacje o procesorze wielordzeniowym jak również możemy się zarejestrować i logować, aby móc przeglądać i porównywać procesory.



Rysunek 4.2: Dalsza część strony głównej.

W dalszej części strony głównej znajdziemy odnośniki do baz procesorów poszczególnych producentów takich jak: Intel, AMD, Sparc, PowerPC i ARM.



Rysunek 4.3: Baza danych.

Schemat bazy danych przedstawia katalog procesorów wielordzeniowych. Pierwszą tabelką o nazwie `t_marki`, która zawiera marki procesorów. Druga tabela o nazwie `t_procesory`, przechowuje dane takie jak: liczbę rdzeni, `max_f_turbo` czyli maksymalną częstotliwość turbo, cache i proces technologiczny. Trzecia tabela nazywa się `uzytkownicy` i zawiera dane takie jak login i hasło.

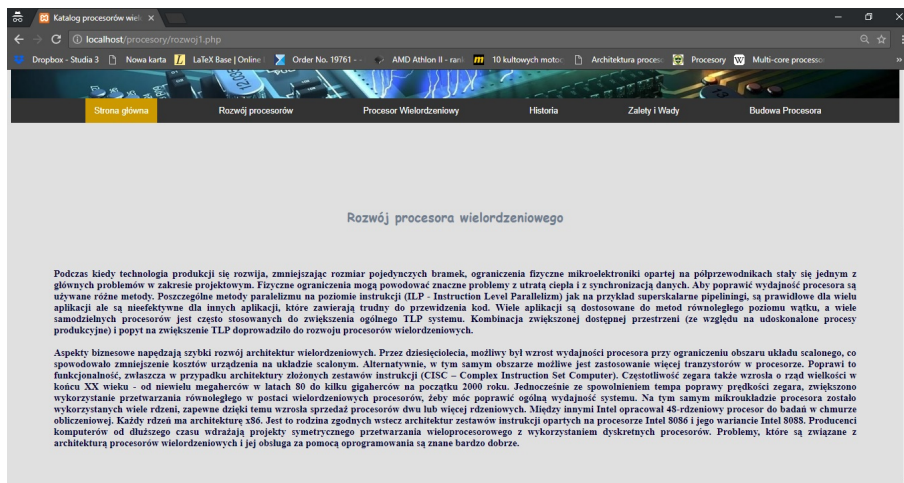


## 4.2 Ogólny opis funkcjonalności dla Gościa



Rysunek 4.4: Menu dla Gościa.

Gość może z paska menu wybrać przycisk: Rozwój procesorów, Procesor Wielordzeniowy, Historia, Zalety i Wady oraz Budowę Procesora i przeczytać informacje.



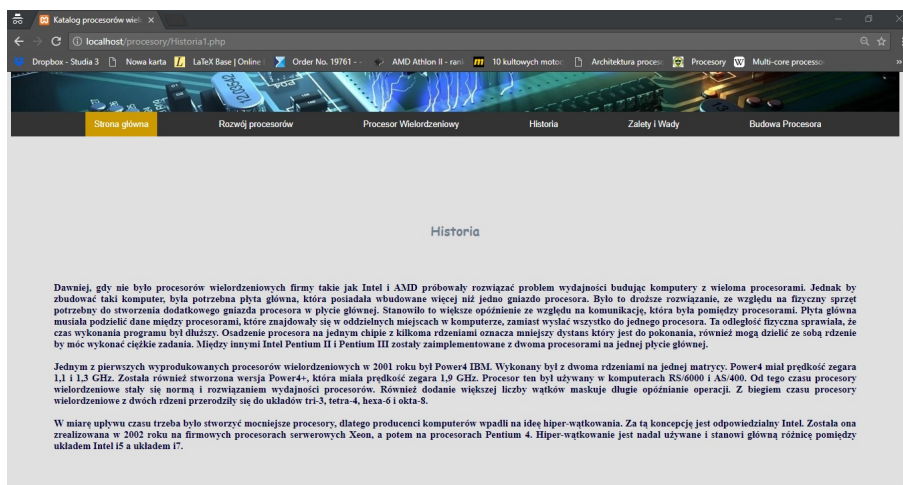
Rysunek 4.5: Rozwój procesora.

Na rysunku 4.5 zamieszczony jest widok podstrony zawierającej wyjaśnienie rozwoju procesora wielordzeniowego.



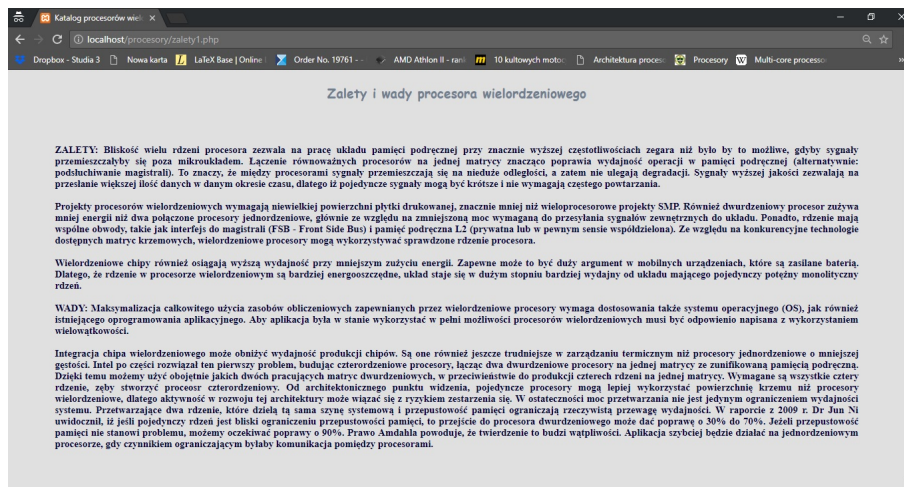
Rysunek 4.6: Pojęcie procesora.

Na rysunku 4.6 zamieszczony jest widok podstrony zawierającej wyjaśnienie pojęcia procesora wielordzeniowego.



Rysunek 4.7: Historia.

Na rysunku 4.7 zamieszczony jest widok podstrony zawierającej wyjaśnienie historii procesora wielordzeniowego.



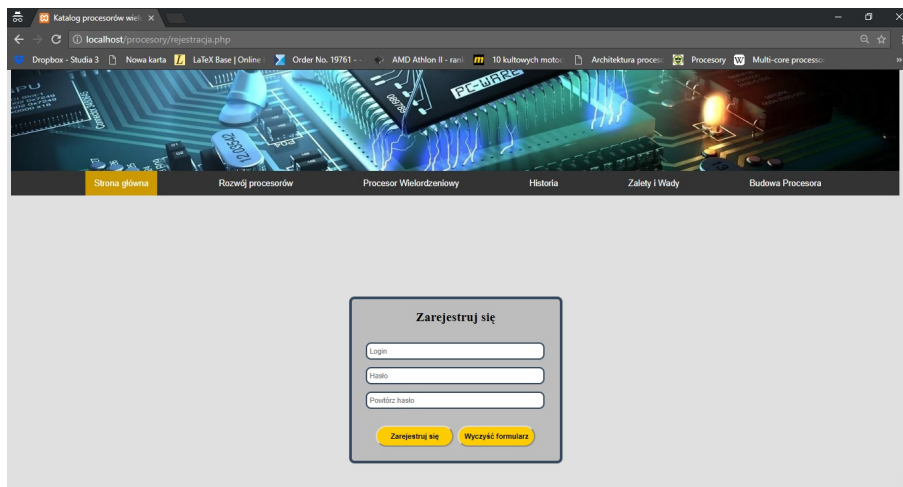
Rysunek 4.8: Zalety i Wady.

Na rysunku 4.8 zamieszczony jest widok podstrony zawierającej wyjaśnienie zalet i wad procesora wielordzeniowego.



Rysunek 4.9: Budowa Procesora.

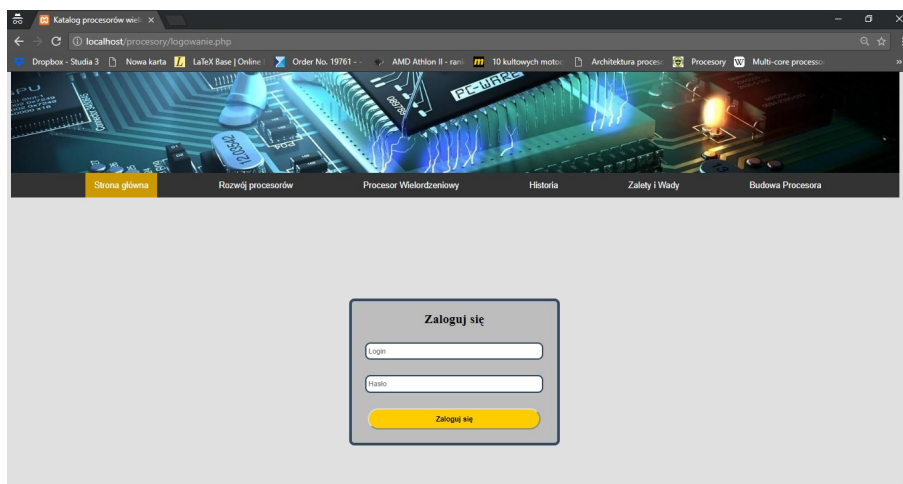
Na rysunku 4.9 zamieszczony jest widok podstrony zawierającej wyjaśnienie budowy procesora wielordzeniowego.



Rysunek 4.10: Rejestracja.

Na rysunku 4.10 znajduje się panel rejestracji dla gościa, po rejestracji przechodzimy do logowania.

### 4.3 Ogólny opis funkcjonalności dla Użytkownika



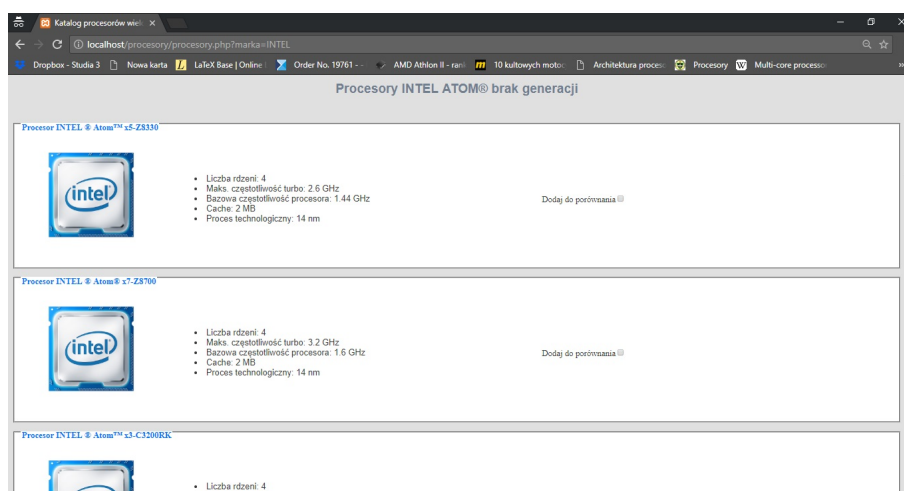
Rysunek 4.11: Logowanie.

Na rysunku 4.11 przedstawiony jest panel logowania dla użytkownika.



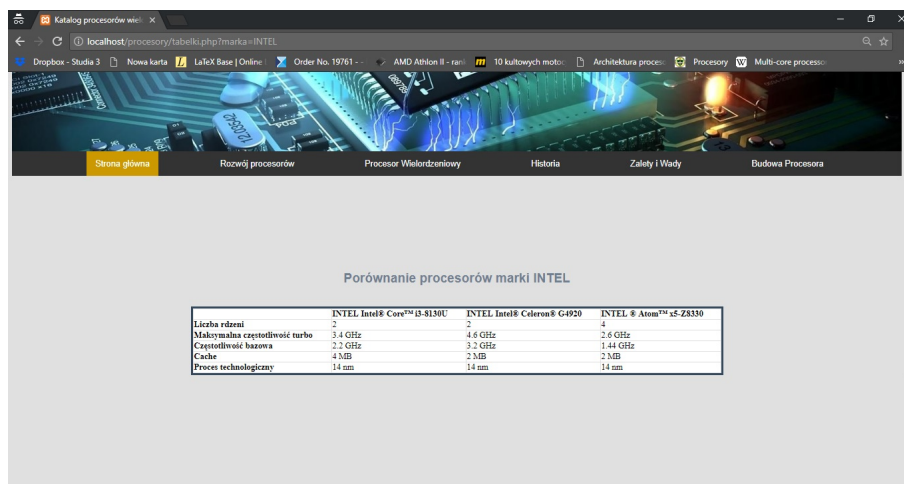
Rysunek 4.12: Panel Użytkownika.

Na rysunku 4.12 znajduje się panel dla użytkownika. Użytkownik posiada takie same prawa jak gość ale dodatkowo ma możliwość porównywania procesorów.



Rysunek 4.13: Katalog procesorów wielordzeniowych.

Na rysunku 4.13 zaprezentowany jest katalog do porównywania procesorów. Należy zaznaczyć minimum dwa a maksymalnie trzy procesory, aby je porównać ze sobą. Po wybraniu danych procesorów klikamy na przycisk porównaj.



	INTEL Intel® Core™ i3-5130U	INTEL Intel® Celeron® G4920	INTEL® Atom™ x5-Z8330
Liczba rdzeni	2	2	4
Maksymalna częstotliwość turbo	3.4 GHz	4.6 GHz	2.6 GHz
Częstotliwość bazowa	2.2 GHz	3.2 GHz	1.44 GHz
Cache	4 MB	2 MB	2 MB
Proces technologiczny	14 nm	14 nm	14 nm

Rysunek 4.14: Porównanie.

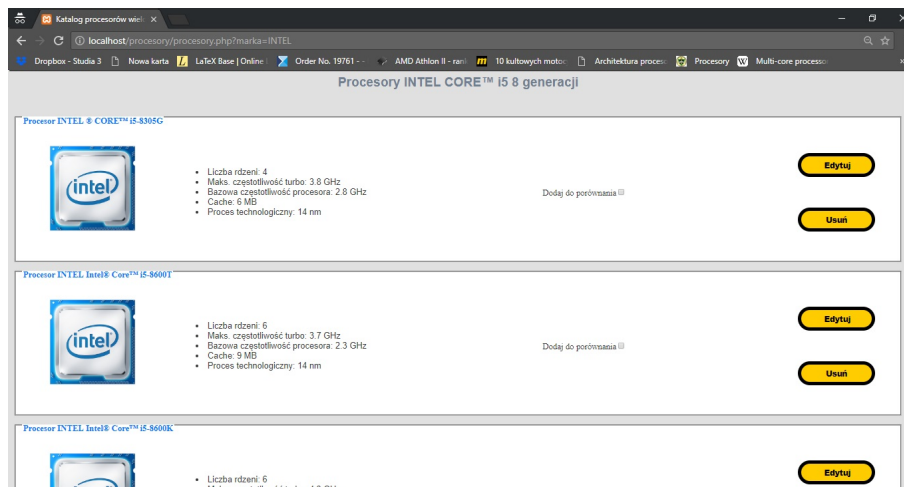
Po wybraniu procesorów i przycisku porównaj przejdziemy do tabelki z porównanymi procesorami.

## 4.4 Ogólny opis funkcjonalności dla Administratora



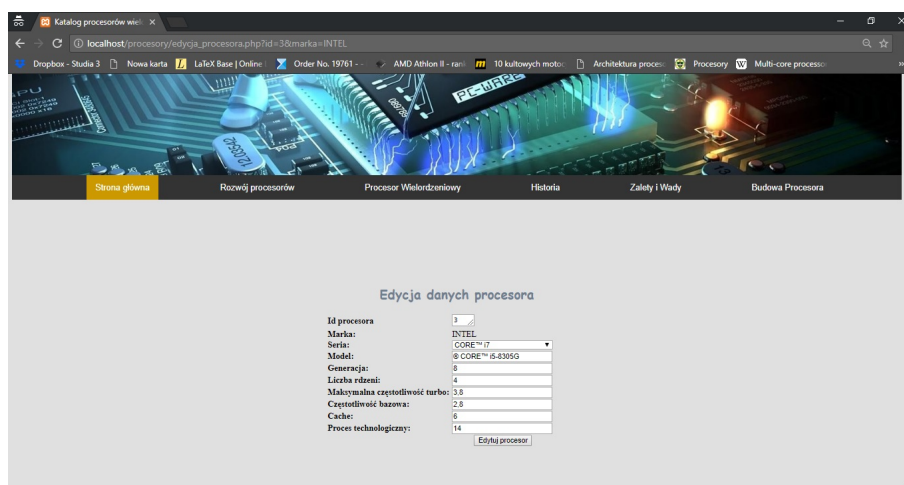
Rysunek 4.15: Logowanie Admina.

Po za logowaniu na profil Admina wyświetla nam się komunikat, że jesteśmy zalogowani jako Admin. Administrator ma takie same prawa jak gość i użytkownik ale dodatkowo może dodawać, edytować i usuwać procesory.



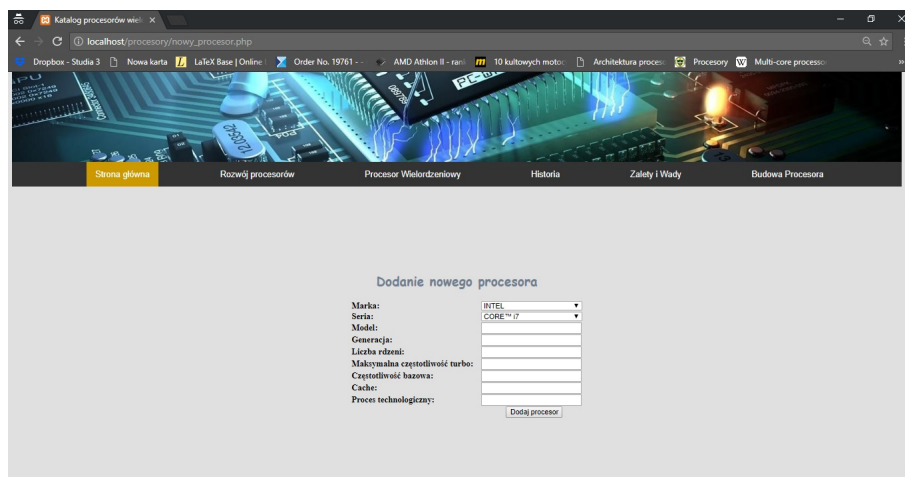
Rysunek 4.16: Panel Admina.

Po przejściu do katalogu np. Procesory wielordzeniowe Intel, Admin może edytować dany procesor lub go usunąć.



Rysunek 4.17: Edycja procesora.

Po kliknięciu edytuj przechodzimy do edycji procesora wielordzeniowego. Wyłącznie Admin może tylko edytować dane procesorów. Po kliknięciu Edytuj procesor zostanie z edytowany i przejdziemy do katalogu w którym edytowaliśmy procesor.



Rysunek 4.18: Dodawanie procesora.

Na rysunku 4.18 jest przedstawiona zakładka dla dodania nowego procesora.



# Słowniczek skrótów

**ARB** - (Address Resolution Buffer) bufor do rozwiązywania problemów.

**big.LITTLE** - heterogoniczna architektura obliczeniowa.

**C2C** - sterownik układu scalonego .

**CISC** - (Complex Instruction Set Computer) architektura procesora, która wykonuje w jednym czasie kilka operacji instrukcji.

**CMP** - (Chip MultiProcessing) wieloprocessorowy układ scalony.

**DIMM** - (Dual In-Line Memory Module) połączony podwójny moduł pamięci.

**DMA** - (Direct Memory Access) bezpośredni dostęp do pamięci.

**DRAM** - (Dynamic Random-Access Memory) pamięć dynamiczna o swobodnym dostępie.

**DSP** - (Digital Signal Processor) cyfrowy procesor sygnałowy czyli do cyfrowej obróbki sygnałów.

**EBC** - kontroler magistrali.

**FPU** - (Floating-Point Unit) jednostka zmiennoprzecinkowa.

**FSB** - (Front Side Bus) magistrala łącząca mikroprocesor.

**GHz** - jednostka częstotliwości.

**GPU** - (Graphics Processing Unit) procesor graficzny, znajduje się na karcie graficznej.

**ILP** - (Instruction Level Parallelizm) równoległość na poziomie instrukcji, jest to miara tego ile można wykonać instrukcji jednocześnie.

**ISA** - (Instruction Set Architecture) architektura zestawu instrukcji.

**L1** - pamięć podręczna pierwszego poziomu.

**L2** - pamięć podręczna drugiego poziomu.

**MIPS** - (Microprocessor without Interlocked Piped Stages) architektura mikroprocesora.

**PCB** - (Printed Circuit Board) płytką drukowaną.

**PGA** - (Pin Grid Array) układ scalony stosowany do pubowy procesorów.

**SMP** - (Symmetric MultiProcessing) symetryczny wieloprocessor.

**SMT** - (Simultaneous Multi-Threading) wykonywanie wielu zadań w czasie rzeczywistym, jednoczesne wielowatkowanie.

**SPU** - (Streaming Processing Units) urządzenie do przetwarzania strumieniowego.

**TLB** - (Translation Lookaside Buffer) bufor pamięci mikroprocesorowej.

**TLP** - (Thread Level Parallelism) równoległość na poziomie wątku, dzielenie programu na niezależne zadania.

**VLIW** - (Very long instruction word) bardzo długie słowo instrukcji [6].

# Podsumowanie

Założeniem mojej pracy licencjackiej było stworzenie aplikacji internetowej: „Katalog procesorów wielordzeniowych”, która zawiera bazę procesorów wielordzeniowych poszczególnych producentów. Aplikacja została wykonana w pełni i można z niej korzystać. Do zrobienia strony zostały użyte technologie programistyczne takie jak: HTML, CSS, MySQL, PHP, JavaScript. W danej aplikacji została zaprojektowana możliwość porównywania procesorów wielordzeniowych znajdujących się w bazie danych. Takie porównanie możliwe jest poza logowaniem się i umożliwia nam zestawienie parametrów wybranych procesorów. Aplikacja zawiera ogólne informacje dotyczące procesorów wielordzeniowych, które są zawarte w poszczególnych podstronach.

W danym systemie zostały stworzone trzy statusy dostępu: *Gość*, *Użytkownik* i *Administrator*. Każdy ze statusów ma przydzielone odpowiednie funkcjonalności korzystania ze strony internetowej.

Aplikacja ma na celu przekazanie użytkownikowi ogólnych informacji o procesorach wielordzeniowych jak również pomóc mu przy wyborze marki i odpowiedniego modelu. Dany system został stworzony z myślą o możliwości rozbudowy w późniejszym czasie poprzez rozszerzenie bazy danych o dodatkowe parametry i wzbogacenie o nowe funkcjonalności.

# Bibliografia

- [1] Stephen W. Keckler · Kunle Olukotun · H. Peter Hofstee Editors „*Multicore Processors and Systems*”, Springer Science+Business Media, LLC 2009.
- [2] Odalys Morrill „*Handbook of CPU Sockets*”, Research World 2012.
- [3] [https://en.wikipedia.org/wiki/Multi-core\\_processor](https://en.wikipedia.org/wiki/Multi-core_processor).  
(dostęp 15-19.03.2018)
- [4] <https://www.burnworld.com/the-history-of-the-multi-core-processor/>  
(dostęp 31.03.2018)
- [5] <https://www.pcmach.com/article/all-about-multi-core-processors-what-they-are-how-they-work-and-where-they-came-from/> (dostęp 31.03.2018)
- [6] [https://pl.wikipedia.org/wiki/Skróty\\_uywane\\_w\\_informatyce](https://pl.wikipedia.org/wiki/Skróty_uywane_w_informatyce)(dostęp 25.05.2018)
- [7] [https://www.purepc.pl/files/Image/news/2015/08/intel\\_core\\_skylake\\_1.jpg](https://www.purepc.pl/files/Image/news/2015/08/intel_core_skylake_1.jpg)