

Najważniejszą cechą charakteryzującą maszynę von Neumanna (w odróżnieniu od innych wczesnych modeli komputerów) jest:

- posiadanie rejestrów
- wykorzystanie licznika rozkazów
- potokowe przetwarzanie rozkazów
- przechowywanie rozkazów i danych w jednej pamięci

Pamięć o dostępie swobodnym definiujemy jako pamięć:

- dostępna dla każdego programu
- o niezależnym bezpośrednim dostępie do każdej komórki
- umożliwiającą dostęp do komórek każdemu procesorowi maszyny
- będącą teoretycznym modelem nieskończonej taśmy z maszyny Turinga

Procesor chcąc uzyskać dostęp do komórki pamięci RAM może posłużyć się:

- symboliczną nazwą komórki
- adresem komórki zapisanym w postaci bitowej
- adresem komórki przechowywanym w rejestrach
- specjalnym rejestrem przechowującym adres w postaci symbolicznej

ALU (jednostka arytmetyczno-logiczna) służy do:

- dekodowania i wykonywania rozkazów
- wykonywania wszystkich rozkazów
- wykonywania m.in. operacji dodawania, mnożenia itp.
- wykonywania m.in. operacji skoku
- wykonywania m.in. operacji pobierania z pamięci RAM

Elementem jednostki centralnej w maszynie von Neumanna nie jest:

- jednostka sterująca
- zegar
- zestaw rejestrów
- pamięć podręczna
- wewnętrzna magistrala
- jednostka arytmetyczno-logiczna

Licznik rozkazów:

- jest to układ liczący wykonane rozkazy
- jest to rejestr zawierający adres kolejnego rozkazu do wykonania
- jest to układ służący do przewidywania liczby rozkazów „omijanych” przy rozkazie skoku
- służy m.in. do realizacji rozkazu skoku

Akumulator (w procesorze) jest to:

- rejestr przechowujący adresy rozkazów do wykonania
- rejestr sumujący wyniki poprzednich operacji
- rejestr przechowujący wyniki dodawania
- rejestr przechowujący dane i wyniki operacji ALU
- rejestr będący domyślnym argumentem rozkazów ALU

Rozkaz pobierany z komórki pamięci przez procesor i następnie wykonywany:

- zawiera nazwę (numer) rozkazu i zawsze wskazania adresów argumentów
- zawiera nazwę (numer) rozkazu i ewentualnie wskazania adresów argumentów (jeśli są rozkazowi potrzebne)
- zawiera nazwę (numer) rozkazu i ewentualnie nazwę (numer) rozkazu zawierającego adresy argumentów
- zawiera tylko nazwę rozkazu, adresy argumentów są w kolejnych komórkach pamięci

Wykonanie każdego rozkazu składa się etapów, wśród których zawsze występują:

- pobranie rozkazu
- zdekodowanie rozkazu
- pobranie argumentów z pamięci
- wykonanie rozkazu
- zapis wyniku do pamięci

Podstawowymi operacjami, koniecznymi do realizacji algorytmów, występującymi na listach rozkazów wszystkich procesorów są:

- operacje arytmetyczne (dodawanie, mnożenie itp.)
- operacje logiczne (koniunkcja, alternatywa, itp.)
- operacje na znakach (odczyt, zapis, zmiana kodowania, itp.)
- operacje na obrazach (odczyt, wyświetlanie, zmiana parametrów itp.)
- operacje na plikach (otwieranie, zamykanie, odczyt, zapis itp.)
- operacje na komórkach pamięci RAM (odczyt zawartości, zapis, itp.)
- operacje skoku

Rozkaz skoku jest realizowany przez procesor poprzez:

- przesunięcie kodu w pamięci operacyjnej
- podmianę aktualnie wykonywanego rozkazu na rozkaz będący celem skoku
- wykonanie procedury systemowej realokacji odniesień do pamięci
- zmianę zawartości rejestru licznika rozkazów, zawierającego adres kolejnego rozkazu do wykonania
- przełączenie kontekstu: zmianę zawartości rejestrów i struktur danych obsługujących aktualnie wykonywany fragment kodu

Maszyna Turinga jest abstrakcyjnym modelem maszyny obliczeniowej, o takich samych możliwościach jak współczesne komputery; charakteryzuje się ją jako maszynę (zaznacz każde właściwe określenie):

- czytającą program z nieskończonej taśmy
- czytającą dane z nieskończonej taśmy
- której działanie polega na zmianach jej stanu
- o skończonej liczbie stanów

Bramka w informatyce to:

- układ elektroniczny realizujący podstawowe jednobitowe operacje logiczne
- układ elektroniczny realizujący podstawowe wielobitowe operacje logiczne
- układ elektroniczny realizujący podstawowe wielobitowe operacje arytmetyczne
- układ elektroniczny służący do zachowania i zmiany stanu jednego ze swoich wyjść poprzez odpowiednie sygnały wejściowe
- układ do przechowania i zmiany pojedynczego bitu

Przerzutnik w informatyce to:

- układ elektroniczny realizujący podstawowe jednobitowe operacje logiczne
- układ elektroniczny realizujący podstawowe wielobitowe operacje logiczne
- układ elektroniczny realizujący podstawowe wielobitowe operacje arytmetyczne
- układ elektroniczny służący do zachowania i zmiany stanu jednego ze swoich wyjść poprzez odpowiednie sygnały wejściowe
- układ do przechowania i zmiany pojedynczego bitu

Funkcja logiczna „p XOR q”, dla kolejnych par wartości p i q równych (0,0), (0,1), (1,0), (1,1) przybiera wartości:

- 0,0,0,0 - 0,1,1,0 1,0,0,1 1,0,0,0 0,0,0,1 1,1,1,0 0,1,1,1

Funkcja logiczna „p NAND q”, dla kolejnych par wartości p i q równych (0,0), (0,1), (1,0), (1,1) przybiera wartości:

- 0,0,0,0 0,1,1,0 1,0,0,1 1,0,0,0 0,0,0,1 1,1,1,0 0,1,1,1

ALU jest to:

- układ elektroniczny realizujący podstawowe jednobitowe operacje logiczne
- układ elektroniczny realizujący podstawowe wielobitowe operacje logiczne
- układ elektroniczny realizujący podstawowe wielobitowe operacje arytmetyczne
- układ elektroniczny służący do zachowania i zmiany stanu jednego ze swoich wyjść poprzez odpowiednie sygnały wejściowe
- układ do przechowania i zmiany pojedynczego bitu
- żadna odpowiedź nie jest prawidłowa

Rejestr jest to:

- układ elektroniczny do przechowywania spisu rozkazów procesora
- układ elektroniczny do przechowywania danych złożony z przerzutników
- podstawowa komórka pamięci wewnętrznej procesora
- układ do przechowywania stanu pamięci podręcznej procesora
- żadna odpowiedź nie jest prawidłowa

Zgodnie z prawem Moore'a liczba tranzystorów umieszczanych w pojedynczym układzie scalonym podwaja się co x miesięcy, gdzie x wynosi:

7 8 17 18 27 28 47 48

Prawo Moore'a stwierdza, że co stałą liczbę (ok. 18) miesięcy podwaja się:

- liczba tranzystorów umieszczanych w pojedynczym układzie scalonym
- wydajność procesorów
- częstotliwość taktowania zegara procesora
- rozmiar pamięci podręcznej procesora
- pojemność twardego dysku

Wskaż prawidłową kolejność etapów przetwarzania potokowego (ID – dekodowanie rozkazu, IE – wykonanie rozkazu, IF – pobranie rozkazu, OF – pobranie argumentów, WB – zapis wyniku):

- IE ID OF IF WB
- ID IF WB IE OF
- IE IF WB ID OF
- WB IE IF ID OF
- IF ID OF IE WB

Potokowe przetwarzanie rozkazów oznacza przetwarzanie:

- potoku rozkazów – kolejny rozkaz po zakończeniu poprzedniego
- zbliżone do pracy na taśmie produkcyjnej – pojedyncza jednostka funkcjonalna procesora realizuje tylko część przetwarzania rozkazu
- dzięki któremu procesor może współbieżnie przetwarzać wiele rozkazów
- wymagające istnienia złożonych procesorów o wielu jednostkach funkcjonalnych
- takie jak w kartach graficznych (inaczej przetwarzanie strumieniowe)

Superskalarność procesora oznacza:

- bardzo rozbudowane potoki przetwarzania (kilkanaście etapów)
- istnienie kilku potoków przetwarzania
- posiadanie więcej niż jednej jednostki funkcjonalnej ALU
- możliwość wykonywania instrukcji poza kolejnością
- wsparcie sprzętowe wielowątkowości

Możliwości równoległości na poziomie wykonania pojedynczego rozkazu (Instruction Level Parallelism) współczesnych procesorów (rdzeni) są ograniczone do następującej liczby rozkazów kończonych w każdym cyklu zegara:

1 kilka kilkanaście kilkadziesiąt kilkaset

Producenci procesorów ogólnego przeznaczenia przestali zwiększać częstotliwość pracy procesorów z powodu:

- zbyt wielu etapów w potokowym przetwarzaniu rozkazów, utrudniających zrównoleglenie kodu
- zbyt wysokiego poziomu wydzielania ciepła
- barier technologicznych w taktowaniu układów elektronicznych
- niemożności zagwarantowania odpowiednio szybkiej pracy pamięci podręcznej
- żadna z wymienionych możliwości

Płyta kuchenki elektrycznej produkuje ok. 10 W/cm² natomiast współczesne procesory około (w W/cm²):

1 10 100 1000 10000

Procesory wielordzeniowe:

- są nazywane inaczej układami scalonymi wieloprocessorowymi
- pracują w modelu SIMD (single instruction multiple data)
- nigdy nie przekroczą liczby rdzeni ok. kilkunastu
- nie posiadają pamięci podręcznej L2 i L3
- żadne stwierdzenie nie jest prawdziwe

Rdzeń procesora wielordzeniowego:

- jest bardzo zbliżony do dawnych procesorów (jednordzeniowych), ale nie potrafi wykonywać bardziej złożonych rozkazów
- jest bardzo zbliżony do dawnych procesorów (jednordzeniowych), ale bez pamięci podręcznej L2 i L3
- wydziela znacznie mniej ciepła niż procesor jednordzeniowy o tej samej częstotliwości pracy
- może mieć własny dostęp do fragmentu pamięci głównej (architektura NUMA)
- żadne stwierdzenie nie jest prawdziwe

W celu realizacji programów procesory zapisują:

- kod i dane w systemie dwójkowym
- kod w systemie szesnastkowym, dane w systemie dwójkowym
- kod w systemie dwójkowym, dane w systemie szesnastkowym
- kod i dane w systemie szesnastkowym

System szesnastkowy zapisu liczb:

- jest sposobem przechowywania liczb przez komputer
- jest używany do zapisu liczb binarnych w formie czytelniejszej dla ludzi
- ma te same cyfry co układ dziesiętny
- ma cyfry: 0, 1, 2, ..., 9, a, b, c, d
- jest wygodny dla programistów, ponieważ jednej cyfrze układu szesnastkowego odpowiadają cztery cyfry układu dwójkowego
- jest układem pozycyjnym, gdzie kolejnym pozycjom (cyfrom) w liczbie przyporządkowane są kolejne potęgi 16 (1, 16, 256, itd.)

Informacja, że procesor jest k-bitowy (np. 32-bitowy) oznacza m.in. że:

- posiada k-bitowe rejestry
- może adresować pamięć od 0 do 2 do potęgi (k-1)
- liczby zmiennoprzecinkowe są przechowywane za pomocą k-bitów
- adresy są przechowywane za pomocą k-bitów
- znaki są przechowywane za pomocą k-bitów

Znaki zapisywane są poprzez podanie związanej z nimi liczby czyli kodu znaku. Liczba znaków jest równa 2 do potęgi k, gdzie k jest liczbą bitów użytych do kodowania znaku. W systemach ASCII i UTF liczby znaków to:

- 64 i 65536
- 2 do potęgi 7 i 65536
- 256 i 2 do potęgi 16
- 2 do potęgi 7 i 2 do potęgi 64
- 256 i 65536

Największą liczbę zapisaną w postaci 32-bitów uzyskuje się przez zastosowanie kodowania jak dla:

- liczb naturalnych
- liczb całkowitych w systemie ze znakiem
- liczb całkowitych w systemie uzupełnień do dwóch
- liczb zmiennoprzecinkowych pojedynczej precyzji
- liczb zmiennoprzecinkowych podwójnej precyzji

Zapis liczby zmiennoprzecinkowej x, polega na zapisie dwóch liczb całkowitych, obliczonych z wykorzystaniem wzoru:

- $x = a/b^2$
- $x = a \cdot 2^b$
- $x = a \cdot 10^b$
- $x = a \cdot \log_2(b)$
- $x = a \cdot \log_{10}(b)$

Sposób kodowania liczb zmiennoprzecinkowych za pomocą skończonej liczby bitów (w praktyce 32 lub 64) oznacza, że m.in.:

- można zapisać tylko skończoną ilość liczb podczas gdy liczb rzeczywistych jest nieskończenie wiele
- na osi liczbowej pomiędzy liczbami zmiennoprzecinkowymi są odstępy (między liczbami rzeczywistymi nie ma, tzn. nie wszystkie liczby rzeczywiste dają się zapisać jako zmiennoprzecinkowe)
- nie wszystkie operacje wykonalne na liczbach rzeczywistych mogą być wykonywane na liczbach zmiennoprzecinkowych
- wynik operacji na liczbach zmiennoprzecinkowych może być liczbą nie dającą się zapisać w postaci zmiennoprzecinkowej
- wyniki operacji na liczbach zmiennoprzecinkowych po zapisaniu w postaci zmiennoprzecinkowej mogą być niedokładne

Big-endian i little-endian to nazwy dwóch różnych sposobów zapisu:

- liczb naturalnych i całkowitych ze względu na liczbę bitów użytych do zapisu
- liczb naturalnych i całkowitych ze względu na kolejność umieszczania bitów w komórkach pamięci
- liczb całkowitych ze względu na sposób uwzględnienia liczb mniejszych od zera
- liczb całkowitych ze względu na liczbę bitów użytych do zapisu

Liczb zmiennoprzecinkowe pojedynczej (32 bity) i podwójnej (64 bity) precyzji różnią się:

- sposobem uwzględnienia znaku liczby
- kolejnością zapisywanych bajtów w komórkach pamięci
- rodzajem rejestrów i jednostek ALU stosowanych do ich obsługi przez procesor
- liczbą bitów użytych do zapisu wykładnika potęgi dwójki
- zakresem czyli największymi i najmniejszymi możliwymi do zapisania liczbami dodatnimi i ujemnymi
- ilością bitów użytych do zapisu liczby przez którą mnożona jest potęga dwójki
- dokładnością czyli odległością dwóch kolejnych liczb

Ze względu na różne sposoby zapisu liczb różnych rodzajów (naturalnych, ułamkowych itp.), a także liczb tego samego rodzaju:

- procesory używają specjalnych adresów dla każdego sposobu zapisu liczb
- w programach wprowadza się pojęcie typu zmiennej i stosuje się zmienne różnych typów
- rozróżnia się systemy 32 i 64 bitowe
- systemy mogą różnić się szczegółami zapisu liczb, nawet liczb tego samego rodzaju (np. naturalnych)
- przy przesyłaniu liczby z jednego systemu do drugiego, każdy z nich może ją odczytać w inny sposób

W programach komputerowych występują zmienne, czyli symbole które mogą przybierać różne wartości.

Analizując wykonanie kodu przez procesor można przyjąć, że:

- zmiennej odpowiada pewien obszar pamięci operacyjnej
- nazwa zmiennej nie ma znaczenia dla procesora
- typ zmiennej nie ma znaczenia dla procesora
- procesor odwołuje się do zmiennej tylko poprzez jej adres w pamięci
- rozmiar zmiennej w pamięci dla danego systemu (np. 32-bitowego) jest taki sam dla zmiennych wszystkich typów

W przypadku wykonania w programie operacji arytmetycznej niedozwolonej lub operacji o wyniku, którego nie da się zapisać w sposób standardowy (np. dzielenie przez zero lub liczbę bardzo bliską zero) zgodnie ze standardami i konwencją system może zgłosić błąd oraz:

- kontynuować działanie programu bez zmian

- kontynuować działanie programu wstawiając specjalne symbole w miejsce wyników
- dokonać korekty programu tak żeby błąd nie wystąpił
- przerwać działanie programu
- przerwać pracę procesora (czyli systemu)

Liczba 1234,5678 w notacji naukowej znormalizowanej, tzn, z jedną cyfrą przed przecinkiem, ma postać:

- $1,2345678 \cdot 10^0$

- $1,2345678 \cdot 10^3$

- $1,2345678 \cdot 10^{-3}$

- $1234567,8 \cdot 10^{-3}$

- $1234567,8 \cdot 10^3$

Język asemblera:

- jest językiem programowania tzw. niskiego poziomu

- jest sposobem zapisu w postaci symbolicznej (łatwiejszej do zrozumienia dla programisty) rozkazów maszynowych procesora

- nie posiada rozkazów skoku (zastąpione przez pętle)

- posługuje się adresami komórek pamięci i nazwami rejestrów zamiast nazw zmiennych

- wprowadza własne nazwy zmiennych zrozumiałe dla procesora

W programach zapisanych w językach programowania wyższego poziomu programiści posługują się zmiennymi, czyli symbolami, które mogą przyjmować różne wartości (podobnie jak w matematyce). W języku asemblera (czyli w rozkazach, które rzeczywiście wykonuje procesor) zamiast tego używa się odniesień do komórek pamięci (wewnętrznej procesora – rejestrów i zewnętrznej RAM) a dostęp do zawartości komórki uzyskuje się podając:

- adres wewnętrzny rejestru w zapisie binarnym, np. 1101001

- nazwę rejestru np. %eax

- adres komórki pamięci w zapisie szesnastkowym np. \$567af3b2

- adres komórki pamięci przechowywany w rejestrze, stosując notację nawiasów oraz podając adres wewnętrzny rejestru, np. (1101001)

- adres komórki pamięci przechowywany w rejestrze, stosując notację nawiasów oraz podając nazwę rejestru np. (%ebx)

Rozkazy języka asemblera, przedstawiające w postaci symbolicznej rozkazy wykonywane przez procesor, zawierają nazwę rozkazu, określającą wykonywaną operację oraz argumenty operacji (jeśli występują), którymi mogą być stałe (liczby, znaki, napisy), zawartości rejestrów, zawartości komórek pamięci oraz:

- etykiety rozkazów,

- adresy rozkazów

- nazwy procedur

- adresy procedur

- nazwy urządzeń wejścia/wyjścia

Zapis (%ecx) w języku asemblera (notacja AT&T) oznacza:

- wartość zmiennej ecx
- adres procedury o nazwie ecx
- zawartość rejestru ecx
- zawartość komórki pamięci o adresie w rejestrze ecx
- adres obiektu ecx

Podstawowymi rozkazami wykonywanymi przez procesor są m.in. rozkazy:

- skoku (bezw warunkowego - jmp i warunkowego – jeq, jl, jge, itp.)
- przeniesienia danych do lub z rejestrów (mov, load, store, itp.)
- operacje logiczne (and, or, xor, itp.)
- operacji arytmetycznych (add, mul, sub, itp.)
- operacje graficzne (draw, clear, itp.)
- obsługi wywołania procedur (call, ret, itp.)

Na stosie w trakcie realizacji programu przechowywane są:

- dane programu
- dane wspólne procedur
- dane wejściowe (argumenty) procedur
- dane prywatne (lokalne) procedur
- dane do komunikacji procedur z systemem operacyjnym
- dane do komunikacji międzyprocesowe

Mechanizm stosu umożliwia efektywną realizację:

- wywołania procedur
- wywołania procedur rekurencyjnych
- komunikacji międzyprocesorowej
- zarządzania przesyłaniem argumentów do procedur
- zarządzania przesyłaniem danych do urządzeń wejścia/wyjścia
- zarządzania użyciem zmiennych lokalnych (pomocniczych) procedur

Użycie procedur niesie ze sobą korzyści w postaci:

- czytelniejszej organizacji kodu
- zmniejszenia rozmiaru kodu
- zwiększenia szybkości działania programu
- ułatwienia modyfikacji i ponownego wykorzystania kodu
- zmniejszenia liczby danych przesyłanych do i z pamięci

W momencie wywołania procedury procesor:

- czyści stos
- przesuwa granice stosu (powiększa stos)
- kopiuje na stos kod wykonywanej procedury
- zapisuje dane na stos (m.in. argumenty wywoływanej procedury)
- przenosi sterowanie w miejsce przechowywania kodu procedury w pamięci operacyjnej (poprzez ustawienie licznika rozkazów)

W momencie powrotu z procedury procesor:

- zapisuje wynik procedury
- kopiuje wartości argumentów ze stosu
- przesuwa granice stosu (zmniejsza stos)
- przenosi sterowanie w miejsce kodu zaraz po wywołaniu procedury

Czas dostępu do komórki pamięci podręcznej jest wielkością rzędu:

- kilku nanosekund
- kilkudziesięciu nanosekund
- kilkuset nanosekund
- kilku mikrosekund
- kilkuset mikrosekund- usuwa kod procedury ze stosu

Czas dostępu do komórki pamięci operacyjnej jest wielkością rzędu:

- kilkudziesięciu pikosekund
- kilku nanosekund
- kilkudziesięciu nanosekund
- kilku mikrosekund
- kilkuset mikrosekund

Pamięć podręczna zorganizowana jest w postaci linii, rozmiar pojedynczej linii jest wielkością rzędu:

- kilku bitów
- kilkudziesięciu bitów
- kilku bajtów
- kilkudziesięciu bajtów
- kilku kilobajtów
- kilkudziesięciu kilobajtów

Rozmiar pamięci podręcznej L1 jest wielkością rzędu:

- kilku bajtów
- kilkudziesięciu bajtów
- kilku kilobajtów
- kilkudziesięciu kilobajtów
- kilkuset kilobajtów
- kilku megabajtów

Procedura dostępu do komórki pamięci operacyjnej w modelu z pamięcią podręczną zawiera następujące operacje (kolejność nie gra roli):

- sprawdzenie komórek pamięci podręcznej odpowiadających komórce pamięci operacyjnej
- w przypadku trafienia, pobranie zawartości z pamięci podręcznej
- w przypadku trafienia modyfikacja stanu linii pamięci podręcznej
- w przypadku chybienia pobranie zawartości z pamięci operacyjnej
- w przypadku chybienia przeładowanie linii pamięci podręcznej i pobranie zawartości z pamięci podręcznej

Linijka pamięci podręcznej:

- służy do szybkiego dostarczenia procesorowi danych, które niedawno były używane przez procesor lub które sąsiadują w pamięci operacyjnej z niedawno używanymi danymi
- służy do poszerzenia zakresu adresowania dzięki rozmiarowi linii rzędu kilkudziesięciu bajtów
- odpowiada jednej liczbie z pamięci operacyjnej
- odpowiada kilku liczbom z pamięci operacyjnej
- jest przeładowywana jeśli procesor nie znajdzie żądanych danych w pamięci podręcznej
- jest przeładowywana przy zapisie danych

Lokalność czasowa odniesień do pamięci oznacza:

- relatywnie rzadkie uzyskiwanie dostępu do pamięci w czasie obliczeń
- uzyskiwanie dostępu do tej samej komórki pamięci w krótkim odstępie czasu
- uzyskiwanie dostępu do sąsiadujących komórek pamięci w krótkim odstępie czasu
- częste odwoływanie się do tej samej zmiennej w programie
- uzyskiwanie wysokiego współczynnika trafień w pamięć podręczną

Lokalność przestrzenna odniesień do pamięci oznacza:

- korzystanie z sąsiednich linii pamięci podręcznej
- uzyskiwanie dostępu do tej samej komórki pamięci w krótkim odstępie czasu
- uzyskiwanie dostępu do sąsiadujących komórek pamięci w krótkim odstępie czasu
- częste odwoływanie się do tej samej zmiennej w programie
- uzyskiwanie wysokiego współczynnika trafień w pamięć podręczną

Pamięć podręczna pozwala na przyspieszenie wykonania programów dzięki temu, że:

- procesor szybciej adresuje komórki pamięci podręcznej niż komórki pamięci operacyjnej
- przesłanie zawartości kilku komórek na raz trwa krócej niż kilkakrotne przesłanie zawartości jednej komórki
- programy wykazują lokalność czasową i przestrzenną
- czas dostępu do komórki pamięci podręcznej jest znacznie krótszy niż czas dostępu do komórki pamięci operacyjnej
- czas przeładowania linijki pamięci podręcznej jest krótszy niż czas dostępu do komórki pamięci operacyjnej

Współczynnik trafień w pamięć podręczną (hit ratio):

- oznacza stosunek liczby żądań dostępu do pamięci zrealizowanych z pamięci podręcznej do liczby wszystkich żądań dostępu do pamięci
- oznacza ile linii pamięci podręcznej procesor musiał sprawdzić zanim znalazł żadaną komórkę
- oznacza ile bloków pamięci operacyjnej przypada na jedną linię pamięci podręcznej
- jego wysoka wartość oznacza że czas dostępu do zmiennych w trakcie realizacji programu jest znacznie krótszy niż gdyby dane pobierać zawsze z pamięci operacyjnej
- jest wysoki jeśli program wykazuje wysoki stopień lokalności czasowej i/lub przestrzennej
- gdy jest wysoki oznacza częste korzystanie z szybkiej pamięci podręcznej zamiast z powolnej pamięci operacyjnej

Pamięć wirtualna

- oznacza całość pamięci dostępnej dla umieszczania zmiennych programu
- oznacza modelową pamięć dla algorytmów nie związaną z wykonywaniem rzeczywistych programów
- jej rozmiar jest najczęściej związany z rozmiarem podstawowych rejestrów procesora
- jest używana żeby przyspieszyć działanie programów korzystających z wielu zmiennych i dużych tablic
- jest używana, żeby ułatwić wykonywanie programów na komputerach z różną ilością pamięci RAM
- oznacza, że pamięć RAM jest przydzielana programowi tylko wtedy kiedy bezpośrednio jej potrzebuje

Błąd strony:

- oznacza błąd w trakcie wykonywania programu, polegający na odwołaniu do niewłaściwego obszaru pamięci
- oznacza odwołanie do adresu na stronie pamięci wirtualnej, która aktualnie nie ma przydzielonej ramki pamięci operacyjnej
- oznacza błąd w adresie strony w pamięci wirtualnej
- powoduje uruchomienie procedury przydziału ramki pamięci operacyjnej dla strony pamięci wirtualnej
- powoduje przerwanie wykonywania programu i zgłoszenie błędu

Procedura obsługi błędu strony:

- oznacza procedurę przydziału ramki pamięci operacyjnej dla strony pamięci wirtualnej, która obecnie nie posiada ramki
- może oznaczać konieczność przepisania zawartości strony pamięci wirtualnej (zapisanej uprzednio na twardym dysku) do ramki pamięci operacyjnej
- jest wewnętrzną procedurą procesora
- jest procedurą systemu operacyjnego uruchamianą automatycznie po błędzie strony
- jest procedurą systemu operacyjnego uruchamianą w ściśle określonych odstępach czasu w celu czyszczenia pamięci operacyjnej

Tablica stron:

- zawiera spis wszystkich stron pamięci wirtualnej
- zawiera spis zawartości stron aktualnie obecnych w pamięci operacyjnej
- zawiera spis numerów ramek pamięci operacyjnej przydzielonych stronom pamięci wirtualnej
- jest częściowo przechowywana w specjalnej szybkiej pamięci podręcznej procesora
- jest w pamięci RAM i jest obsługiwana przez specjalny kontroler pamięci

Magistrala:

- jest inaczej nazywana szyną danych
- jest jedna w komputerze, przesyła dane do procesora
- służy do przesyłania danych między różnymi urządzeniami w komputerze
- w najnowszych wersjach, przy łączeniu procesorów z pamięcią ma przepustowości rzędu kilku MB/s
- w najnowszych wersjach, przy łączeniu procesorów z pamięcią ma przepustowości rzędu kilkuset MB/s
- w najnowszych wersjach, przy łączeniu procesorów z pamięcią ma przepustowości rzędu kilkudziesięciu GB/s

Spośród rozmaitych magistral w komputerze rozróżniamy magistrale łączące:

- procesory z pamięcią
- procesory z kartami rozszerzeń (np. PCI Express)
- komputery ze sobą w sieć (np. Ethernet)
- procesory z różnymi urządzeniami zewnętrznymi (np. USB)
- procesory z twardymi dyskami (np. SATA)
- komputery z urządzeniami przenośnymi (np. EDGE, WiFi)

Pamięć EPROM:

- jest nowoczesnym odpowiednikiem twardego dysku
- jest małą, nieulotną pamięcią
- służy do sterowania pracą urządzeń komputera
- służy do przechowania BIOSu
- jest używana podczas uruchamiania systemu

BIOS:

- jest elementem standardowych systemów operacyjnych
- jest małym wewnętrznym systemem operacyjnym komputera
- jest używany podczas pracy komputera do obsługi urządzeń
- jest używany do uruchamiania systemu
- służy do znalezienia podstawowych urządzeń komputera i zawartego na nich oprogramowania standardowego systemu operacyjnego
- dostarcza oprogramowania (sterowników) urządzeń plug-and-play

Nazwa SMP oznacza systemy:

- jednoprocessorowe
- wieloprocessorowe, tzw. symetryczne (każdy procesor widzi system tak samo)
- jednoprocessorowe z kartami graficznymi używanymi do obliczeń ogólnego przeznaczenia
- z jednolitym dostępem do pamięci UMA (zawartość każdej komórki pamięci operacyjnej dostarczana procesorowi w takim samym czasie)
- z niejednolitym dostępem do pamięci NUMA (zawartości różnych komórek pamięci operacyjnej mogą być dostarczane procesorowi w różnych czasach)

Nazwa NUMA oznacza systemy wieloprocessorowe (wielordzeniowe):

- z jednolitym dostępem do pamięci (zawartość każdej komórki pamięci operacyjnej dostarczana procesorowi w takim samym czasie)
- z niejednolitym dostępem do pamięci (zawartości różnych komórek pamięci operacyjnej mogą być dostarczane procesorowi w różnych czasach)
- skalujące się (dające się praktycznie wykorzystywać) dla liczb procesorów do rzędu kilkunastu
- skalujące się (dające się praktycznie wykorzystywać) dla liczb procesorów do rzędu kilkudziesięciu
- skalujące się (dające się praktycznie wykorzystywać) dla liczb procesorów do rzędu kilkuset

Klastry:

- są specjalistycznymi superkomputerami
- powstają przez połączenie wielu komputerów siecią i wyposażenie ich w specjalne oprogramowanie pozwalające traktować je jak pojedynczy system do uruchamiania programów
- zakładają najczęściej model programowania bez pamięci wspólnej
- są najdroższymi komputerami równoległymi
- skalują się (dają się praktycznie wykorzystywać) dla liczb procesorów (rdzeni) do rzędu kilkudziesięciu
- skalują się (dają się praktycznie wykorzystywać) dla liczb procesorów (rdzeni) do rzędu kilku tysięcy
- skalują się (dają się praktycznie wykorzystywać) dla liczb procesorów (rdzeni) do rzędu kilkuset tysięcy

Spośród poniższych urządzeń komputera NIE SĄ niezbędne:

- procesor
- pamięć RAM
- chipset
- magistrala
- twardy dysk
- pamięć nieulotna (np. EPROM)
- karta sieciowa

Cechą architektury harwardzkiej jest:

- a) przechowywanie zmiennych w pamięci wspólnie z kodem programu,
- b) rozdzielanie pamięci danych od pamięci programu,
- c) przechowywanie zmiennych w pamięci dynamicznej, a kodu programu w pamięci statycznej,
- d) posiadanie oddzielnych magistral: pamięci danych i pamięci algorytmu.

Określić stan flag Sign (znak), Carry (przeniesienie), Zero i Overflow (nadmiar) po wykonaniu działania $(-7)+5$ (argumenty czterobitowe, kodowanie U2).

S=1 C=1 Z=0 O=0

Przesunięcie zawartości rejestru eax w lewo o dwa bity odpowiada:

- a) mnożeniu wartości w eax przez 2,
- b) mnożeniu przez 4 zawartości komórki pamięci wskazanej przez wartość w rejestrze,
- c) mnożeniu wartości w rejestrze przez 4,
- d) dzieleniu wartości w rejestrze przez 4.

Cechą pamięci SDRAM jest:

- a) cykliczne odświeżanie zawartości,
- b) podział na banki,
- c) podział adresu na adres rzędu i kolumny,
- d) przechowywanie zawartości po wyłączeniu zasilania,
- e) możliwość łatwej alokacji pamięci.

Licznik rozkazów procesora (PC-program counter, IP instruction pointer) jest to rejestr służący do:

- a) odmierzania czasu wykonywania danych procedury przez procesor,
- b) wyznaczania adresu skoku względnego,
- c) generowania przerwania co określoną liczbę wykonywanych instrukcji,
- d) przechowywania adresu kolejnej instrukcji do pobrania i wykonania.

Stosując kod uzupełnień do dwóch (U2), w zmiennej ośmiobitowej można poprawnie zapisać wartość z zakresu:

- a) 0-255
- b) -128-127
- c) -127-128

Liczbę zmiennoprzecinkową R można zapisać za pomocą liczb całkowitych: S-znak, M-mantysa-ułamek, W-wykładnik, P-podstawa systemu liczbowego powiązanych zależnością:

- a) $R = S * M * P^W$
- b) $R = P * M * S^W$
- c) $R = S * M * \log P(W)$

Jaki typ zależności w kodzie rozwiązuje technika przemianowania rejestrów (register renaming):

- a) WaR b) RaW
- c) WaW d) WaR i WaW
- e) WaR i RaW f) wszystkie kombinacje powyższych

Korzyści odnoszone z obecności pamięci podręcznej cache w systemie komputerowym związane są z:

- a) lokalnością odniesień w kodzie programu, odpowiednią organizacją danych w pamięci RAM komputera,
- b) lokalnością czasową – częstotliwością odwołań do określonych danych i kodu programu,
- c) czasem dostępu do głównej pamięci operacyjnej,
- d) mapowaniem portów I/O w przestrzeni adresowej procesora

Cykl rozkazowy procesora składa się następującej sekwencji cykli maszynowych:

- a) pobranie instrukcji -> inkrementacja licznika rozkazów -> dekodowanie informacji -> wykonanie operacji -> zapis wyniku operacji i/lub ustawienie flag
- b) pobranie instrukcji -> dekodowanie instrukcji -> ustawienie flag -> wykonanie operacji -> zapis wyniku
- c) pobranie instrukcji -> dekodowanie instrukcji -> wykonanie operacji -> zapis wyniku operacji i/lub ustawienie flag -> inkrementacja licznika rozkazów
- d) pobranie instrukcji -> dekodowanie instrukcji -> wykonanie operacji -> zapis wyniku operacji/ustawienie flag -> reset licznika rozkazów

Z punktu widzenia teorii układów cyfrowych procesor jest układem:

- a) sekwencyjnym,
- b) proporcjonalnym,
- c) kombinacyjnym,
- d) liniowym

Przerzutnik może być traktowany jako:

- a) podstawowy element funkcjonalny pamięci rejestru mogący przechowywać najmniejszą jednostkę informacji,
- b) układ przełączający magistrale: danych i adresową,
- c) program obsługi kilku monitorów ekranowych w środowisku graficznym,
- d) element portu wyjściowego,
- e) element portu wejściowego.

Algorytm Write-Back:

- a) jest stosowany w obsłudze pamięci wirtualnej,
- b) nie jest stosowany w obsłudze pamięci cache,
- c) jest stosowany w kontrolerze przerwań,
- d) powoduje potencjalne problemy ze spójnością danych w pamięci cache i RAM,

Obsługa zdarzenia typu błąd strony (page fault) odbywa się przez:

- a) zgłoszenie przerwania niemaskowalnego
- b) zgłoszenie wewnętrznego przerwania CPU- wyjątku,
- c) polling
- d) użycie kanału DMA
- b) o przesłania brakującej strony.

Najnowsze procesory zgodne z x86 (i5, i7 itp)

- a) posiadają zintegrowany kontroler pamięci,
- b) posiadają wiele jednostek wykonawczych o zróżnicowanej funkcjonalności,
- c) mogą równolegle wykonywać kilkanaście instrukcji (w obrębie jednego rdzenia),
- d) posiadają pamięć cache L3 odrębną dla każdego rdzenia,
- e) posiadają zintegrowany mostek północny.

W architekturze von Neumana:

- a) komórki pamięci mają adresy będące wyłącznie wielokrotnością podstawy użytego systemu liczbowego,
- b) bez analizy zawartości pamięci nie można jednoznacznie określić, czy dany blok pamięci zawiera kod programu, czy dane,
- c) zmienne przechowywane są w komórkach pamięci o adresach parzystych, a kod programu – nieparzystych,
- d) zorganizowana jest pamięć operacyjna komputerów PC.

Określić stan flag Sign (znak), Carry (przeniesienie), Zero i Overflow (nadmiar) po wykonaniu działania $(-5)+7$ (argumenty czterobitowe, kodowanie U2).

S=0 C=0 Z=0 O=0

Pamięć podręczna cache L1:

- a) zoptymalizowana jest pod kątem minimalizacji czasu dostępu (CPU do danych w pamięci),
- b) wykorzystuje pełną skojarzeniowość (odzworowanie typu fully associative),
- c) wykorzystuje najczęściej algorytm write-through,
- d) jest współdzielona przez rdzenie procesora.

Wykonanie rozkazu xor eax, eax da w wyniku:

- a) $eax=0$ i flaga $Z=1$
- b) $eax=1$ i flaga $Z=0$
- c) $eax=0$ i flaga $Z=0$
- d) wynik będzie zależny od wartości początkowej rejestru eax .

Procesory CISC cechuje:

- a) wiele różnych trybów adresowania,
- b) prosta realizacja przetwarzania potokowego – dzięki dużej liczbie złożonych i specjalizowanych instrukcji,
- c) obecność instrukcji, upraszczających ręczne programowanie w języku assembler,
- d) bardzo mały pobór prądu.

System przerwań zgodny z architekturą x86 obejmuje:

- a) przerwania chronione i wirtualne,
- b) przerwania niemaskowalne,
- c) przerwania sprzętowe: zewnętrzne i wewnętrzne (wyjątki) oraz programowe,
- d) przerwania modulowane fazą sygnały zegarowego,

Liczbę zmiennoprzecinkową R można zapisać za pomocą liczb całkowitych: S-znak, M-mantysa-ułamek, W-wykładnik, P-podstawa systemu liczbowego powiązanych zależnością:

- a) $R = S * M * \log P(W)$
- b) $R = P * M * S^W$
- c) $R = S * M * P^W$

Funkcje systemowe w Linuxie (system calls) uruchamia się wywołując przerwanie:

- a) programowe 80h
- b) programowe 21h
- c) sprzętowe 80h
- d) sprzętowe 21h

Procesory ARM/Cores(?) stosowane obecnie na szeroką skalę np. w procesorach do urządzeń przenośnych mają architekturę typu:

- a) RISC
- b) CISC
- c) SPARC
- d) CISC przekształconą wewnętrznie do RISC

Jednostka arytmetyczno-logiczna (ALU) w procesorze służy do:

- a) wykonywania operacji m.in.: $*$, $+$, $-$, and, or, add, (?)
- b) steruje przełączaniem bloków procesora między wykonywaniem instrukcji arytmetycznych i logicznych,
- c) może obliczać komórek pamięci i sterować wykonywaniem skoków warunkowych.

DMA (Direct Memory Access) wprowadzono żeby:

- a) procesor nie musiał czynnie zajmować się transferem danych (I/O-pamięć),
- b) maksymalizować przepustowość urządzenia I/O pamięć,
- c) poprawić efektywność działania systemu przerwań,
- d) poprawić efektywność działania pamięci cache,
- e) system efektywniej przełączał kontekst procesora.

Z przetwarzaniem potokowym wiążą się:

- a) hazardy danych,
- b) konflikty w dostępie do różnych jednostek wykonawczych procesora,
- c) utrata spójności pamięci cache,
- d) spekulacyjne wykonywanie instrukcji warunkowych/skoków,
- e) uruchomienie procesów systemowych i użytkownika w dwóch oddzielnych potokach.

Problemem występującym w zapisie binarnym liczb wykorzystującym bit znaku (np. float, znak-moduł) jest występowanie:

- a) podwójnego zera (+0 i -0),
- b) NaN (Not a Number),
- c) symboli nieoznaczonych,
- d) wyjątków zgłaszanych przez CPU

Wykonując rozkaz `mov eax,[var]` (składnia Intel), do rejestru `eax` wczytana zostanie:

- a) wartość zmiennej `var`,
- b) adres, pod którym umieszczona jest zmienna `var`,
- c) adres segmentu pamięci, w którym umieszczona jest zmienna `var`,
- d) wartość przechowywana w obszarze pamięci rozpoczynającym się od adresu określonego wartością `var`

Mechanizm pamięci wirtualnej jest wspierany przez:

- system operacyjny
- CPU
- system operacyjny i CPU
- TLB (Translation Look-aside Buffer)

W jakim trybie działa procesor komputera PC (zgodny z x86) podczas pracy w systemie Linux?

- chronionym
- rzeczywistym
- w celu zapewnienia zgodności, emuluje procesory z systemów unixowych

Czynniki ograniczające możliwości dalszego zwiększania częstotliwości taktowania CPU to:

- długość pojedynczej fazy w przetwarzaniu potokowym
- problemy z synchronizacją CPU z urządzeniami I/O
- problemy z odprowadzaniem ciepła z struktury CPU
- długość połączeń między blokami CPU

Instrukcją przekształcającą wartość `11001100B` w 8-bitowym rejestrze `R` na `11000000B` jest:

```
Sar R,4  
shl R,4  
xor R,0000001100B  
or R,1100000000B  
and R,11110000B
```

Która przestrzeń adresowa jest większa:

- z reguły wirtualna
- z reguły fizyczna
- mogą być równe

Cztery komórki pamięci o adresach od 00 do 03 zawierają kolejno bajty: 12 34 AB CD. Jaka liczba zapisana jest w pamięci, zakładając, że system pracuje w porządku little-endian? (wszystkie liczby podano w porządku szesnastkowym)

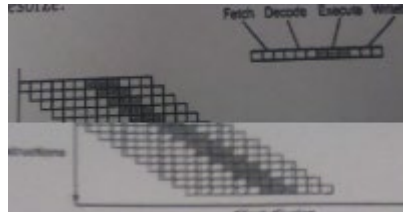
1234ABCD

CDAB3412

2143BADC

DCBA4321

Poniższy schemat obrazuje wykonywanie rozkazów w procesorze:



superskalarnym

superpotokowym

posiadającym kilka potoków

posiadającym 3 poziomy pamięci cache

posiadającym 3 jednostki ALU

Procesory jednordzeniowe, superskalarne, mogą teoretycznie przetwarzać równolegle 4 instrukcje (is... width=4), w praktyce wykonują w jednym cyklu:

-więcej niż cztery instrukcje jednego wątku

-cztery instrukcje jednego wątku

-ok. dwóch instrukcji jednego wątku,

-mogą uzupełniać luki w potokach instrukcjami i wątku

Wykonanie rozkazu skoku warunkowego: jne etykieta/adres (jump_if_not_equal) możliwe będzie wtedy gdy flaga:

-Z (zero)=0

-Z (zero)=1

-S (sign)=1

-Wynik ostatniej operacji był liczbą nieparzystą

Stos:

-umożliwia przekazywanie argumentów do wywoływanej funkcji

-zawiera adresy fizyczne stron pamięci

-jest buforem typu LIFO (last in, first out),

-jest buforem typu FIFO (first in, first out).

W zmiennej ośmiobitowej ze znakiem (np. char, kodowanie U2) nie można poprawnie zapisać liczby całkowitej z zakresu:

-128+127

0+255

-127+128

Korzyści ze stosowania pamięci cache mogą zostać zwiększone poprzez:

- wzrost częstotliwości odwołań do pliku wymiany,
- częstsze przełączanie kontekstu procesorów
- odpowiednie rozmieszczenie danych w pamięci operacyjnej

We współczesnych procesorach zgodnych z x86, np. Intel i5, i7:

- Istnieje możliwość wykonania kilku różnego typu instrukcji równolegle (w obrębie jednego zadania)
- Instrukcje x86 są tłumaczone na sekwencje prostych rozkazów (mikrokod),
- Sieć połączeń międzynarodowych jest zrealizowana w topologii hiperkostki,
- Zdekodowana instrukcja x86 zostaje przesłana do jednej z kilkunastu jednostek wykonawczych
- rdzenie nie współdzielą bloku pamięci cache L3

Pamięć typu flash:

- traci zawartość po odłączeniu zasilania
- posiada ograniczoną liczbę cykli programowania/kasowania
- stosowana jest do budowy rejestrów procesora
- powinna przechowywać zapisaną informację przez okres od kilku do kilkunastu lat(w prawidłowo zaprojektowanym i eksploatowanym urządzeniu),
- w jednej komórce możliwe jest przechowanie informacji o stanie więcej niż jednego bitu

Struktury CMOS i MOSFETT kojarzą się z:

- interfejsami komunikacyjnymi
- technologią budowy współczesnych cyfrowych układów scalonych
- językami programowania

Tablica stron (page table):

- jest elementem mechanizmu pamięci wirtualnej
- nie jest umieszczona w pamięci operacyjnej komputera,
- jest strukturą danych przeglądarki internetowej
- nie jest strukturą zawierającą informacje o wykorzystaniu pamięci cache,
- jest strukturą, którą posiada każdy proces działający pod kontrolą wielozadaniowego systemu operacyjnego(obsługującego mechanizm pamięci wirtualnej).

1. Przejście z eax = 0xFF00... na jakiś inny

- xor eax, 0xFFFF...
- shl +8

2. Instrukcja xor ax, ax

- ax = 0, Z=1
- znaczy to samo co and ax0

3. Architektura harwardzka:

- wykorzystywana przy cache
- oddzielenie pamięci danych od pamięci programu, czy tam oddzielne magistrale

ALE: fakt że kod w pamięci statycznej, a dane w dynamicznej jest ZŁĄ ODPOWIEDZIĄ, poprawne były tylko dwie

4. Korzyści z pamięci cache:

- odniesienie do RAM
- lokalność (chyba odniesień, generalnie ta dłuższa odpowiedź z lokalnością)

5. Stos:

- jest buforem typu LIFO //nie jest FIFO!
- wskaźnik na ostatni odłożony element

6. Biblioteki dynamiczne:

- coś z dobrym zagospodarowaniem pamięcią
- że występuje to lazy binding

7. Kojarzy mi się coś o działania typu set associative, ale nie wiem o co chodziło już

- miało być że jest w L1
- coś z jakimś podziałem

8. Zadanie z kodem : miało wyjść chyba 10 i 22, nie pytajcie mnie dlaczego

9. Do czego NIE JEST wykorzystywany licznik rozkazów:

-

Nie pamiętam odpowiedzi więc napiszę te, które były **BŁĘDNE** (czyli gdyby teraz było do czego służy to to):

- przechowywania adresu kolejnej instrukcji do wykonania
- wyznaczania adresu skoku względnego

10. Co nie wpływa na coś tam (to gdzie były odpowiedzi z WaW, RaW itp.), poprawne były:

- RaW i chyba jeszcze coś (generalnie te na R)

11. Procesory (x86):

- działają w trybie chronionym w Linuxie
- ring 0

Najnowsze procesory zgodne z x86:

- postrzegane przez programistów jako CISC
- mogą obsługiwać 2-3 operacje jednocześnie na jednym rdzeniu

12. Wywołanie call:

- skok bezwarunkowy (jedyna dobra odpowiedź)

Wykonanie rozkazu „and eax0” daje wyniku:

- a) Eax=0 i flaga Z=1
- b) Eax=1 i flaga Z=0
- c) Eax bez zmian stan flagi Z będzie zależny od wartości rejestru eax
- d) Eax=0 i flaga Z=0
- e) Eax=1 i flaga Z=1

Po operacji arytmetycznej (lub logicznej) wykonanie rozkazu skoku warunkowego „je etykieta” (jump if equal) możliwe tylko przy następującym ustawieniu flag:

- a) Z=0 i S=0
- b) Z=1 i S=1
- c) Z=1 i S=0
- d) Z=0 i S=1

Uszeregować interfejsy zaczynając od mającego największą przepustowość(1) do najwolniejszego

3 USB 2.0

1 PCI Express x16 3.0

4 PS/2

2 seria ata

W architekturze x86 adresy portów I/O znajdują się w:

- a) Przestrzeni adresowej pamięci operacyjnej
- b) Przestrzeni adresowej urządzeń I/O
- c) Przestrzeni adresowej pamięci ROM
- d) W tablicy deskryptorów portów
- e) W dwóch, niezależnych przestrzeniach – osobno dla urządzeń I oraz O

Podać liczbę wszystkich zależności danego typu, występujących w kodzie (składnia rozkazów MIPS: operacja, wynik_operacji, argument1, argument2)

1 add r1,r2,3 0 write after write

2 xor r2,r3,r1 2 write after read

3 and r3,r1,2 2 read after write

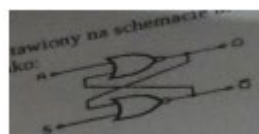
FMA to:

- a) Poprawione DMA (fast memory access)
- b) Poprawione zarządzanie pamięcią (fast memory allocation)
- c) Wykonywanie działań $a*b+c$ przy pomocy jednej instrukcji
- d) Zwiększenie dokładności poprzez redukcję liczby zaokrągleń pośrednich wyników w działaniu $a*b+c$ (fused multiply-add)

Komputer osobisty z wielordzeniowym procesorem (np.i7) może być uważany za:

- a) System wieloprocesorowy o pamięci rozproszonej
- b) System wieloprocesorowy o pamięci wspólnej
- c) Komputer, w którym dla każdego rdzenia dostępna jest cała przestrzeń adresowa
- d) Komputer z wieloma procesorami uniwersalnymi, dostosowanymi do wykonywania różnych zadań
- e) Komputer, w którym czas dostępu do pamięci operacyjnej zależy adresu konkretnej komórki

Układ przedstawiony na schemacie może znaleźć zastosowanie jako:



- a) Element rejestru
- b) Element porty wyjściowego
- c) Element portu wejściowego
- d) Element pamięci dynamicznej RAM
- e) Element pamięci statycznej RAM

Bramka realizująca funkcję eXOR może znaleźć zastosowanie:

- a) Przy generowaniu/sprawdzaniu bitu parzystości
- b) Jako „sterowana” zewnętrznym sygnałem bramka negacji
- c) W komparatorze (dekoderze równości argumentów)
- d) W generowaniu flagi overflow
- e) W generowaniu flagi Carry(przeniesienie)
- f) W sprawdzeniu poprawności działania arytmetycznych w kodzie U2

Instrukcja przekształcająca wartość 01010101B w 8-bitowym rejestrze R na 10101010B jest:

- a) Shr R,1
- b) SarR,1
- c) xorR,10101010B
- d) xorR, 00000000B
- e) xorR, 11111111B
- f) shlR,1

Instrukcje push i pop (w CPU zgodnych z x86):

- a) Związane są z obsługą stosu
- b) Związane są z transferem danych do/z pamięci
- c) Modyfikują wartość wskaźnika stosu
- d) Modyfikują wartość ramki stosu
- e) Modyfikują znaczniki procesora

W procesorach Pentium4, Core2, i7 zastosowano wielowątkowość typu:

- a) Fine-grained
- b) Coarse-grained,
- c) SMT(simultaneous MultiThreading)

W komputerze z pamięcią cache działającą w oparciu o algorytm write-through:

- a) Gwarantowana jest maksymalna wydajność systemu
- b) Każdy zapis z pamięci cache jest przenoszony do pamięci RAM w celu utrzymania spójności
- c) Dane z procesora do pamięci cache są przesyłane przez dodatkowy bufor
- d) Praca pamięci cache jest zależna od konfiguracji pamięci wirtualnej

Cztery komórki pamięci o adresach od 00 do 03 zawierają kolejno bajty: 12 34 AA CD. Jaka liczba zapisana jest w pamięci, zakładając, że system pracuje w porządku big-endian?

- a) CDAB3412
- b) 1234ABCD
- c) 2143BADC
- d) DCBA432

Licznik rozkazów (PC-Program Counter lub IP-Instruction Pointer) służy do:

Pytanie 1
Zakończono
Punkty: 5,00
1" Offaguj pytanie

Licznik rozkazów (PC-Program Counter lub IP-Instruction Pointer) służy do:

Wybierz jedną lub więcej:

- ☒ a. przechowywania adresu kolejnej instrukcji do pobrania i wykonania
- ☐ b. pomiarów wydajności, optymalizacji i profilowania kodu
- ☐ c. wyznaczania adresu skoku (względnej) w x86-64
- ☐ d. zliczania wykonanych rozkazów od ostatniego resetu procesora
- ☐ e. pomiaru czasu wykonywanego programu

A

Zakres zmienności typu signed int (32 bity):

Pytanie 2
Zakończono
Punkty: 5,00
1" Offaguj pytanie

Zakres zmienności typu signed int (32 bity):

Wybierz jedną lub więcej:

- ☐ a. jest taki sam jak zakres typu float (32 bity)
- ☐ b. jest większy od zakresu typu float (32 bity)
- ☒ c. nie jest większy od zakresu typu double (64 bity)
- ☒ d. jest mniejszy od zakresu typu float (32 bity)

C, D

Rezultat funkcji (Linux 64 bit) będący liczbą całkowitą powinien zostać zwrócony w rejestrze:

Pytanie 3
Zakończono
Punkty: 5,00
1" Offaguj pytanie

Rezultat funkcji (Linux 64 bit) będący liczbą całkowitą powinien zostać zwrócony w rejestrze:

Wybierz jedną lub więcej:

- ☐ a. %rdi
- ☐ b. %rcx
- ☐ c. %rbx
- ☒ d. %rax

D

%ecx=3. Która z poniższych operacji spowoduje, że w tym rejestrze znajdzie się dowolna liczba ujemna (kod U2)?

Pytanie 4
Zakończono
Punkty: 5,00
1" Offaguj pytanie

%ecx=3. Która z poniższych operacji spowoduje, że w tym rejestrze znajdzie się dowolna liczba ujemna (kod U2)?

Wybierz jedną lub więcej:

- ☐ a. ror \$2,%ecx
- ☒ b. neg %ecx
- ☐ c. rol \$4,%ecx
- ☒ d. shl \$29,%ecx
- ☐ e. ror \$1,%ecx
- ☐ f. shl \$31,%ecx
- ☐ g. not %ecx

A, B, E, F, G

Cechą pamięci typu SDRAM-DDR jest:

Pytanie 5
Zakończono
Punkty: 5,00
1" Offaguj pytanie

Cechą pamięci typu SDRAM-DDR jest:

Wybierz jedną lub więcej:

- ☐ a. możliwość pracy z przeplotem (interleaved memory access)
- ☐ b. możliwość przesyłania danych podczas rosnącego i opadającego zbocza sygnału zegarowego
- ☐ c. budowa komórek pamięci z przerzutników RS
- ☐ d. konieczność cyklicznego odświeżania zawartości
- ☐ e. przechowywanie zawartości po wyłączeniu zasilania
- ☒ f. podział pamięci na banki

A, B, D, F

Stałe używane w kodzie programu poprzedzamy znakiem (składnia AT&T):

Pytanie 6
Zakończono
Punkty: 5.00
❏ Oflaguj pytanie

Stałe używane w kodzie programu poprzedzamy znakiem (składnia AT&T):

Wybierz jedną lub więcej:

- ☒ a. \$
- ☐ b. &
- ☐ c. *
- ☐ d. @

A

Standardowe rozszerzenie plików z kodem źródłowym w asemblerze GAS to:

Pytanie 7
Zakończono
Punkty: 5.00
❏ Oflaguj pytanie

Standardowe rozszerzenie plików z kodem źródłowym w asemblerze GAS to:

Wybierz jedną lub więcej:

- ☐ a. *.a
- ☒ b. *.s
- ☒ c. *.as
- ☒ d. *.asm

B

Instrukcja CALL (w x86):

Pytanie 8
Zakończono
Punkty: 5.00
❏ Oflaguj pytanie

Instrukcja CALL (w x86):

Wybierz jedną lub więcej:

- ☒ a. wymaga dostępu do pamięci
- ☒ b. jest rodzajem skoku bezwarunkowego
- ☐ c. jest rodzajem skoku warunkowego
- ☐ d. wykonuje się zależnie od układu flag arytmetycznych
- ☐ e. może spowodować przepełnienie stosu
- ☒ f. wywołuje procedury System Calls w Linuxie

A, B

W jednym rejestrze wektorowym %xmm można umieścić:

Pytanie 9
Zakończono
Punkty: 5.00
❏ Oflaguj pytanie

W jednym rejestrze wektorowym %xmm można umieścić:

Wybierz jedną lub więcej:

- ☐ a. cztery zmienne podwójnej precyzji
- ☐ b. osiem zmiennych pojedynczej precyzji
- ☒ c. cztery zmienne pojedynczej precyzji
- ☒ d. dwie zmienne podwójnej precyzji
- ☐ e. 16 bajtów

C, D, E

Mechanizm pamięci wirtualnej nie wykorzystuje:

Pytanie 10
Zakończono
Punkty: 5.00
❏ Oflaguj pytanie

Mechanizm pamięci wirtualnej nie wykorzystuje:

Wybierz jedną lub więcej:

- ☒ a. algorytmu Write-Through
- ☐ b. tablicy stron (Page Table)
- ☐ c. algorytmu Write-Back
- ☐ d. cache-bufora TLB (Translation Lookaside Buffer)

A

Numer funkcji systemowej w systemie Linux (64 bit) przekazuje się w rejestrze:

Pytanie 11
Zakończono
Punkty: 5,00
Offlaguj pytanie

Numer funkcji systemowej w systemie Linux (64 bit) przekazuje się w rejestrze:

Wybierz jedną lub więcej:

- ☐ a. %rdi
- ☐ b. %rdx
- ☐ c. %rsi
- ☐ d. %rbp
- ☒ e. %rax
- ☐ f. %rcx
- ☐ g. %rbx
- ☐ h. %rsp

E

Przesunięcie logiczne zawartości ośmiobitowego rejestru w prawo o cztery bity odpowiada:

Pytanie 12
Zakończono
Punkty: 5,00
Offlaguj pytanie

Przesunięcie logiczne zawartości ośmiobitowego rejestru w prawo o cztery bity odpowiada:

Wybierz jedną lub więcej:

- ☐ a. rotacji o 4 bity w prawo
- ☒ b. podzieleniu wartości w rejestrze przez 16 (bez znaku)
- ☐ c. przesunięciu arytmetycznemu wartości w rejestrze o 4 bity w lewo
- ☐ d. w przypadku liczby dodatniej (kod U2) - przesunięciu arytmetycznemu wartości w rejestrze o 4 bity w prawo
- ☐ e. rotacji o 4 bity w lewo
- ☐ f. pomnożeniu wartości w rejestrze przez 16 (bez znaku)

B, D

Rozkazem przekształcającym 32-bitową wartość `eax=0xFFFF0000` w `0x0000FFFF` jest:

Rozkazem przekształcającym 32-bitową wartość `eax=0xFFFF0000` w `0x0000FFFF` jest:

Wybierz jedną lub więcej:

- ☐ a. `shr $16,%eax`
- ☒ b. `not %eax`
- ☒ c. `rol $16,%eax`
- ☐ d. `neg %eax`
- ☒ e. `ror $16,%eax`
- ☐ f. `or $0xFFFFFFFF,%eax`
- ☐ g. `xor $0xFFFFFFFF,%eax`
- ☐ h. `sar $16,%eax`

A, B, C, E, G

Instrukcja „LOOP label” zmienia zawartość rejestru:

Instrukcja "LOOP label" zmienia zawartość rejestru:

Wybierz jedną lub więcej:

- ☒ a. %ecx
- ☐ b. %ebx
- ☐ c. %edx
- ☐ d. %eax

A

Parametry funkcji (Linux 64 bitowy) można przekazać:

Parametry funkcji (Linux 64 bitowy) można przekazać:

Wybierz jedną lub więcej:

- ☒ a. w rejestrach ogólnego przeznaczenia (rax)
- ☐ b. w rejestrach jednostki x87
- ☒ c. w rejestrach wektorowych jednostki SSE
- ☒ d. na stosie

A, C

Nazwy rejestrów w składni AT&T poprzedzamy znakiem:

Nazwy rejestrów w składni AT&T poprzedzamy znakiem:

Wybierz jedną lub więcej:

- ☐ a. \$
- ☐ b. &
- ☒ c. %
- ☐ d. ^

C

Mając tablicę zdefiniowaną jako:

Mając tablicę zdefiniowaną jako:

table: .long 7,2,8,6,2

po wykonaniu rozkazów:

xor %esi,%esi

inc %esi

mov \$table,%rdi

liczba "2" zostanie pobrana z tablicy do %edx instrukcją o składni:

Wybierz jedną lub więcej:

- ☐ a. mov (%rdi,%rsi,4),%edx
- ☐ b. mov (%rsi,%rdi,4),%edx
- ☒ c. mov %rdi(%rsi,4),%edx
- ☐ d. mov table(%rsi,4),%edx

C, D

W których instrukcjach konieczny jest sufix (przyrostek) określający wielkość/typ danych:

Pytanie 7

Zakończone

Punkty: 5,00

🚩 Oflaguj pytanie

W których instrukcjach konieczny jest sufix (przyrostek) określający wielkość/typ danych:

Wybierz jedną lub więcej:

- ☐ a. decl %edi
- ☐ b. cmpb \$5, x
- ☐ c. movq %eax, result
- ☒ d. movb \$10, %al
- ☐ e. incl counter

D? (chuj wie)

Organizacja pamięci cache N-way Set Associative:

Pytanie 8

Zakończone

Punkty: 5,00

🚩 Oflaguj pytanie

Organizacja pamięci cache N-way Set Associative:

Wybierz jedną lub więcej:

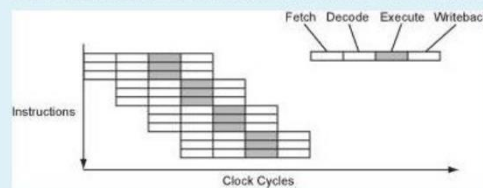
- ☒ a. zawsze zapewnia najwyższy możliwy współczynnik trafień
- ☐ b. stanowi kompromis między % trafień, a komplikacją układu
- ☐ c. podział na sekcje ogranicza jej użycie do architektury harwardzkiej
- ☒ d. jest wykorzystywana w pamięci L1
- ☐ e. z powodu podziału na sekcje nie ma konieczności użycia bloku TAG
- ☐ f. nie jest wykorzystywana w pamięci L3 (współdzielonej między rdzeniami)

B, D

Poniższy schemat obrazuje wykonywanie rozkazów w idealnym, hipotetycznym procesorze:

Pytanie 11
Zakończone
Punkty: 5,00
Oflaguj pytanie

Poniższy schemat obrazuje wykonywanie rozkazów w idealnym, hipotetycznym procesorze:



Wybierz jedną lub więcej:

- ☒ a. rozpoczynającym i kończącym jedną instrukcję w każdym cyklu maszynowym (po wypełnieniu całego potoku)
- ☐ b. posiadającym trójpoziomą pamięć cache (L1-L3)
- ☐ c. wykonującym jedną instrukcję w jednym takcie zegara
- ☐ d. posiadającym co najmniej 3 jednostki wykonawcze
- ☒ e. posiadającym 4 potoki
- ☒ f. superskalarnym

A, D, E, F

Wywołana funkcja (Linux 64 bit) nie może zmieniać zawartości rejestrów:

Pytanie 1
Zakończone
Punkty: 5,00
Oflaguj pytanie

Wywołana funkcja (Linux 64 bit) nie może zmienić zawartości rejestrów:

Wybierz jedną lub więcej:

- ☒ a. %rbp
- ☒ b. %rsp
- ☐ c. %rcx
- ☒ d. %rbx
- ☐ e. %rdx
- ☐ f. %rax

A, B, D

FPU x87:

Pytanie 3
Zakończone
Punkty: 5,00
Oflaguj pytanie

FPU x87:

Wybierz jedną lub więcej:

- ☐ a. zawsze wykonuje obliczenia na liczbach typu double extended precision (80 bitowych)
- ☐ b. zawsze przechowuje dane (argumenty i wyniki) w rejestrach 80 bitowych
- ☒ c. umożliwia ustawienie żądanej precyzji obliczeń
- ☐ d. umożliwia bezpośrednie przenoszenie danych między rejestrami %st(i) i rejestrami ogólnego przeznaczenia %rax

B, C

Stos:

Pytanie 6
Zakończone
Punkty: 5,00
Oflaguj pytanie

Stos:

Wybierz jedną lub więcej:

- ☒ a. wykorzystano do organizacji pracy rejestrów w FPU x87
- ☐ b. nie jest buforem typu LIFO (Last In, First Out)
- ☒ c. nie jest buforem typu FIFO (First In, First Out)
- ☐ d. umożliwia przekazywanie argumentów do wywołanej funkcji
- ☐ e. posiada wskaźnik zawierający adres ostatnio odfiltrowanego na stos elementu (w x86)
- ☐ f. zawiera adresy fizyczne stron pamięci

A, C, D?, E

Problem „podwójnego zera” (+0, -0) dotyczy:

Pytanie **2**

Zakończone

Punkty: 5,00

🚩 Oflaguj pytanie

Problem "podwójnego zera" (+0, -0) dotyczy:

Wybierz jedną lub więcej:

- ☒ a. typów danych o pojedynczej i podwójnej precyzji
- ☐ b. tablicy ASCII
- ☐ c. kodu uzupełnień do dwóch (U2)
- ☐ d. kodu Graya

A

Stosując kod uzupełnień do dwóch (U2) ośmiobitowa zmienna ze znakiem może przyjąć wartość z zakresu:

Pytanie **4**

Zakończone

Punkty: 5,00

🚩 Oflaguj pytanie

Stosując kod uzupełnień do dwóch (U2) ośmiobitowa zmienna ze znakiem może przyjąć wartość z zakresu:

Wybierz jedną lub więcej:

- ☐ a. $-127 \div 128$
- ☐ b. $0 \div 255$
- ☒ c. $-128 \div 127$

C

Jedna linia pamięci cache we współczesnym procesorze x86-64 ma długość:

Pytanie **7**

Zakończone

Punkty: 5,00

🚩 Oflaguj pytanie

Jedna linia pamięci cache we współczesnym procesorze x86-64 ma długość:

Wybierz jedną lub więcej:

- ☐ a. 64 bitów
- ☐ b. 512 bitów
- ☐ c. równą stronie pamięci wirtualnej
- ☒ d. 64 bajtów
- ☐ e. długość linii cache zależy od długości rejestrów wektorowych, a tym samym od mikroarchitektury i generacji procesora

D

Które rejestry nie są wykorzystywane do przekazywania parametrów funkcji (będących liczbami całkowitymi) w systemie Linux (64 bit):

Pytanie 8

Zakończone

Punkty: 5,00

🚩 Oflaguj pytanie

Które rejestry nie są wykorzystywane do przekazywania parametrów funkcji (będących liczbami całkowitymi) w systemie Linux (64 bit):

Wybierz jedną lub więcej:

- ☒ a. %rsp
- ☒ b. %rax
- ☐ c. %rsi
- ☒ d. %rbx
- ☐ e. %r8
- ☒ f. %rbp
- ☐ g. %r9
- ☒ h. %r10
- ☐ i. %rcx
- ☐ j. %rdi
- ☐ k. %rdx

A, B, F, H,

W Linuxie (64 bit) funkcje systemowe wywołuje się instrukcją:

Pytanie 9

Zakończone

Punkty: 5,00

🚩 Oflaguj pytanie

W Linuxie (64 bit) funkcje systemowe wywołuje się instrukcją:

Wybierz jedną lub więcej:

- ☐ a. CALL
- ☐ b. SYSJMP
- ☐ c. INT 80
- ☒ d. SYSCALL
- ☐ e. JMP

D

Użycie bibliotek dynamicznych:

Pytanie 11

Zakończone

Punkty: 5,00

🚩 Oflaguj pytanie

Użycie bibliotek dynamicznych:

Wybierz jedną lub więcej:

- ☐ a. wymusza użycie adresu absolutnego komórki jako argumentu rozkazu wywołującego funkcje biblioteczną
- ☒ b. pozwala zmniejszyć zapotrzebowanie na wolną pamięć RAM (w porównaniu do wersji statycznej)
- ☐ c. gwarantuje najszybsze wywoływanie procedur z bibliotek
- ☒ d. pozwala na zastąpienie funkcji bibliotecznych nowszymi wersjami bez konieczności ponownej kompilacji programu głównego

B

Pamięć cache L1 we współczesnych procesorach x86-64:

Pytanie 12
Zakończono
Punkty: 5,00
Oflaguj pytanie

Pamięć cache L1 we współczesnych procesorach x86-64:

Wybierz jedną lub więcej:

- ☒ a. jest zoptymalizowana pod kątem najkrótszego czasu dostępu (również najkrótszego czasu sprawdzenia, czy żądane dane są obecne w L1)
- ☒ b. zaprojektowana jest zgodnie z architekturą harwardzką
- ☐ c. jest wspólna dla wszystkich rdzeni procesora
- ☐ d. zaprojektowana jest zgodnie z architekturą von Neumanna
- ☒ e. może zawierać dane znajdujące się na stosie
- ☒ f. zazwyczaj nie posiada pojemności kilku-kilkunastu megabajtów
- ☒ g. podzielona jest na bloki danych i instrukcji

A, B, E?, G

Wykonanie rozkazu XOR %eax, %eax:

Wykonanie rozkazu: XOR %eax,%eax:

Wybierz jedną lub więcej:

- ☐ a. jest równoważne operacji: AND \$0,%eax
- ☐ b. jest równoważne operacji OR \$0,%eax
- ☒ c. daje wynik: %eax=0 i flaga Z=1
- ☐ d. daje wynik liczbowy równoważny operacji: MOV \$0,%eax
- ☐ e. daje wynik: %eax=0 i flaga Z=0
- ☐ f. daje wynik zależny od wartości początkowej rejestru

A, C, D,

Po wykonaniu sekwencji operacji na stosie w x86-64:

Po wykonaniu sekwencji operacji na stosie w x86-64:

```
push    %rbp
push    %rbx
push    %rax
pop      %rbx
pop      %rax
pop      %rbp
```

Wybierz jedną lub więcej:

- ☐ a. wskaźnik stosu został przesunięty o ramkę stosu (rbp)
- ☒ b. wartości w rejestrach nie uległy zmianie
- ☒ c. wartość wskaźnika stosu nie uległa zmianie
- ☐ d. stan rejestrów odpowiada wykonaniu rozkazu xchg %rax,%rbx (eXCHanGe)

C, D

W arytmetyce komputerowej (system dwójkowy / kod U2) odejmowanie $A - B$ realizuje się poprzez dodanie do A:

W arytmetyce komputerowej (system dwójkowy / kod U2) odejmowanie $A - B$ realizuje się poprzez dodanie do A:

Wybierz jedną lub więcej:

- ☐ a. zanegowanej bitowo wartości B ($\neg B$)
- ☐ b. B z zanegowanym najstarszym bitem
- ☒ c. uzupełnienia liczby B obliczanego jako: $\neg B + 1$
- ☐ d. B z zanegowaną flagą CARRY

C

Pytanie 3
Zakończono
Punkty: 5,00
🚩 Oflaguj pytanie

W systemie Linux (64 bit) funkcja może zwrócić rezultat będący liczbą zmiennoprzecinkową:

Wybierz wszystkie poprawne:

- ☐ a. w parze rejestrów edx : eax
- ☒ b. w rejestrze ogólnego przeznaczenia %rax
- ☐ c. na stosie
- ☐ d. w rejestrze wektorowym jednostki SSE

Pytanie 4
Zakończono
Punkty: 5,00
🚩 Oflaguj pytanie

Problem "podwójnego zera" (+0, -0) dotyczy:

Wybierz wszystkie poprawne:

- ☐ a. tablicy ASCII
- ☐ b. kodu uzupełnień do dwóch (U2)
- ☒ c. typów danych o pojedynczej i podwójnej precyzji
- ☐ d. kodu Graya

Pytanie 5
Zakończono
Punkty: 5,00
🚩 Oflaguj pytanie

Użycie bibliotek dynamicznych:

Wybierz wszystkie poprawne:

- ☒ a. pozwala na zastąpienie funkcji bibliotecznych nowszymi wersjami bez konieczności ponownej kompilacji programu głównego
- ☐ b. gwarantuje najszybsze wywoływanie procedur z bibliotek
- ☐ c. wymusza użycie adresu absolutnego komórki jako argumentu rozkazu wywołującego funkcję biblioteczną
- ☐ d. pozwala zmniejszyć zapotrzebowanie na wolną pamięć RAM (w porównaniu do wersji statycznej)

W systemie Linux (64bit) funkcja może zwrócić rezultat będący liczbą zmiennoprzecinkową:

D mi w assemblerze wyszło że tak średnio od tego są imo właśnie %xmm)

C (jeżeli stos FPU też się liczy)

Problem „podwójnego zera” (+0,-0) dotyczy:

C

https://en.wikipedia.org/wiki/Signed_zero

Użycie bibliotek dynamicznych:

A i D

Pytanie 6
Zakończzone
Punkty: 5,00
Oflaguj pytanie

Mając tablicę zdefiniowaną jako:

table: .long 7,2,8,6,2

po wykonaniu rozkazów:

```
xor %esi,%esi  
inc %esi  
mov $table,%rdi
```

liczba "2" zostanie pobrana z tablicy do %edx instrukcją o składni:

Wybierz wszystkie poprawne:

- ☐ a. mov %rdi,(%rsi,4),%edx
- ☐ b. mov (%rdi,%rsi,4),%edx
- ☒ c. mov table,(%rsi,4),%edx
- ☐ d. mov (%rsi,%rdi,4),%edx

Pytanie 7
Zakończzone
Punkty: 5,00
Oflaguj pytanie

Skok warunkowy JE (Jump if Equal) wykonany zostanie ZAWSZE po instrukcji:

Wybierz wszystkie poprawne:

- ☒ a. sub %rbx,%rbx
- ☐ b. xor %rax,%eax
- ☐ c. sub %edx,%rdx
- ☐ d. xor \$0,%eax
- ☒ e. xor %eax,%eax
- ☐ f. and %ecx,%ecx
- ☒ g. and \$0,%ebx

Mając tablicę zdefiniowaną:

Table: .long 7,2,8,6,2

B, C

Skok warunkowy JE (Jump if Equal) wykonany zostanie ZAWSZE po instrukcji:

A, D, E, G

Pytanie 1
Zakończzone
Punkty: 5,00
Oflaguj pytanie

Stosując kod uzupełnień do dwóch (U2) ośmiobitowa zmienna ze znakiem może przyjąć wartość z zakresu:

Wybierz wszystkie poprawne:

- ☐ a. 0+255
- ☒ b. -128+127
- ☐ c. -127+128

Pytanie 2
Zakończzone
Punkty: 5,00
Oflaguj pytanie

Które rejestry nie są wykorzystywane do przekazywania parametrów funkcji (będących liczbami całkowitymi) w systemie Linux (64 bit):

Wybierz wszystkie poprawne:

- ☒ a. %rsp
- ☐ b. %rdx
- ☐ c. %rsi
- ☒ d. %rbx
- ☒ e. %rbp
- ☒ f. %r10
- ☒ g. %rdi
- ☒ h. %rax
- ☐ i. %r9
- ☐ j. %r8
- ☐ k. %rcx

Stosując kod uzupełnień do dwóch (U2)

B

Które rejestry nie są wykorzystywane do przekazywania:

A, D, E, F, H

Figure 3.4: Register Usage

Register	Usage	Preserved across function calls
%rax	temporary register; with variable arguments passes information about the number of vector registers used; 1 st return register	No
%rbx	callee-saved register; optionally used as base pointer	Yes
%rcx	used to pass 4 th integer argument to functions	No
%rdx	used to pass 3 rd argument to functions; 2 nd return register	No
%rsp	stack pointer	Yes
%rbp	callee-saved register; optionally used as frame pointer	Yes
%rsi	used to pass 2 nd argument to functions	No
%rdi	used to pass 1 st argument to functions	No
%r8	used to pass 5 th argument to functions	No
%r9	used to pass 6 th argument to functions	No
%r10	temporary register, used for passing a function's static chain pointer	No
%r11	temporary register	No
%r12-r15	callee-saved registers	Yes
%xmm0-%xmm1	used to pass and return floating point arguments	No
%xmm2-%xmm7	used to pass floating point arguments	No
%xmm8-%xmm15	temporary registers	No
%ymm0-%ymm7	temporary registers	No
%st0,%st1	temporary registers; used to return long double arguments	No
%st2-%st7	temporary registers	No
%fs	Reserved for system (as thread specific data register)	No
mxcsr	SSE2 control and status word	partial
x87 SW	x87 status word	No
x87 CW	x87 control word	Yes

Pytanie 8
Zakończone
Punkty: 5,00
🚩 Oflaguj pytanie

Po wykonaniu sekwencji operacji na stosie w x86-64:

```

push    %rbp
push    %rbx
push    %rax
pop     %rbx
pop     %rax
pop     %rbp

```

Wybierz wszystkie poprawne:

- ☐ a. stan rejestrów odpowiada wykonaniu rozkazu xchg %rax,%rbx (eXCHanGe)
- ☒ b. wartość wskaźnika stosu nie uległa zmianie
- ☐ c. wskaźnik stosu został przesunięty o ramkę stosu (rbp)
- ☐ d. wartości w rejestrach nie uległy zmianie

Pytanie 9
Zakończone
Punkty: 5,00
🚩 Oflaguj pytanie

Znakiem komentarza w assemblerze GAS jest:

Wybierz wszystkie poprawne:

- ☐ a. %
- ☐ b. ?
- ☐ c. @
- ☒ d. #

Po wykonaniu sekwencji operacji na stosie w X86-64:

A,B

Znakiem komentarza w assemblerze GAS jest:

D

Pytanie 10
Zakończone
Punkty: 5,00
🚩 Oflaguj pytanie

Cechą pamięci typu SDRAM-DDR jest:

Wybierz wszystkie poprawne:

- ☐ a. możliwość przesyłania danych podczas rosnącego i opadającego zbocza sygnału zegarowego
- ☐ b. przechowywanie zawartości po wyłączeniu zasilania
- ☒ c. konieczność cyklicznego odświeżania zawartości
- ☒ d. podział pamięci na banki
- ☐ e. budowa komórek pamięci z przerzutników RS
- ☐ f. możliwość pracy z przeplotem (interleaved memory access)

Pytanie 11
Zakończone
Punkty: 5,00
🚩 Oflaguj pytanie

Wykonanie rozkazu: XOR %eax,%eax:

Wybierz wszystkie poprawne:

- ☒ a. daje wynik: %eax=0 i flaga Z=1
- ☐ b. daje wynik: %eax=0 i flaga Z=0
- ☒ c. jest równoważne operacji: AND \$0,%eax
- ☐ d. jest równoważne operacji OR \$0,%eax
- ☒ e. daje wynik liczbowy równoważny operacji: MOV \$0,%eax
- ☐ f. daje wynik zależny od wartości początkowej rejestru

Cechą pamięci typu SDRAM-DDR jest:

A, C, D, Potwierdź ktoś co z E (chyba nie) i F (no idea)

Wykonanie rozkazu: XOR %eax,%eax:

A, C, E

Pytanie 12
Zakończone
Punkty: 5,00
Oflaguj pytanie

W Linuxie (64 bit) funkcje systemowe wywołuje się instrukcją:

Wybierz wszystkie poprawne:

- ☐ a. INT 80
- ☒ b. SYSCALL
- ☐ c. JMP
- ☐ d. SYSJMP
- ☐ e. CALL

W Linuxie (64bit) funkcje systemowe wywołuje się instrukcją:

B

Pytanie 10
Zakończone
Punkty: 5,00
Oflaguj pytanie

Wywoływana funkcja (Linux 64 bit) nie może zmienić zawartości rejestrów:

Wybierz jedną lub więcej:

- ☐ a. %rcx
- ☒ b. %rbp
- ☐ c. %rax
- ☒ d. %rsp
- ☐ e. %rdx
- ☒ f. %rbx

Wywołana funkcja (Linux 64bit) nie może zmienić zawartości rejestrów:

B,D,F

Register	Usage	Preserved across function calls
%rax	temporary register; with variable arguments passes information about the number of vector registers used; 1 st return register	No
%rbx	callee-saved register; optionally used as base pointer	Yes
%rcx	used to pass 4 th integer argument to functions	No
%rdx	used to pass 3 rd argument to functions; 2 nd return register	No
%rsp	stack pointer	Yes
%rbp	callee-saved register; optionally used as frame pointer	Yes
%rsi	used to pass 2 nd argument to functions	No
%rdi	used to pass 1 st argument to functions	No
%r8	used to pass 5 th argument to functions	No
%r9	used to pass 6 th argument to functions	No
%r10	temporary register, used for passing a function's static chain pointer	No
%r11	temporary register	No
%r12-r15	callee-saved registers	Yes
%xmm0-%xmm1	used to pass and return floating point arguments	No
%xmm2-%xmm7	used to pass floating point arguments	No
%xmm8-%xmm15	temporary registers	No
%mmx0-%mmx7	temporary registers	No
%st0,%st1	temporary registers; used to return long double arguments	No
%st2-%st7	temporary registers	No
%fs	Reserved for system (as thread specific data register)	No
mxcsr	SSE2 control and status word	partial
x87 SW	x87 status word	No
x87 CW	x87 control word	Yes

Pytanie 11

Zakończone

Punkty: 5,00

🚩 Oflaguj pytanie

Rozkazem przekształcającym 32-bitową wartość `eax=0xFFFF0000` w `0x0000FFFF` jest:

Wybierz jedną lub więcej:

- ☐ a. `ror $16,%eax`
- ☐ b. `or $0xFFFFFFFF,%eax`
- ☐ c. `shr $16,%eax`
- ☐ d. `sar $16,%eax`
- ☐ e. `rol $16,%eax`
- ☒ f. `xor $0xFFFFFFFF,%eax`
- ☐ g. `not %eax`
- ☐ h. `neg %eax`

Pytanie 12

Zakończone

Punkty: 5,00

🚩 Oflaguj pytanie

Rezultat funkcji (Linux 64 bit) będący liczbą całkowitą powinien zostać zwrócony w rejestrze:

Wybierz jedną lub więcej:

- ☐ a. `%rcx`
- ☐ b. `%rbx`
- ☐ c. `%rdx`
- ☒ d. `%rax`

Rozkazem przekształcającym 32bitową wartość `eax=0xFFFF0000` w `0x0000FFFF` jest:

C, F, G

Rezultat funkcji (Linux 64bit) będący liczbą całkowitą powinien zostać zwrócony w rejestrze:

D,C

Pytanie 1

Zakończone

Punkty: 5,00

🚩 Oflaguj pytanie

Standardowe rozszerzenie plików z kodem źródłowym w assemblerze GAS to:

Wybierz jedną lub więcej:

- ☐ a. `*.as`
- ☒ b. `*.asm`
- ☒ c. `*.s`
- ☐ d. `*.a`

Pytanie 2

Zakończone

Punkty: 5,00

🚩 Oflaguj pytanie

Wykonanie rozkazu: `XOR %eax,%eax`:

Wybierz jedną lub więcej:

- ☐ a. daje wynik: `%eax=0` i flaga `Z=0`
- ☐ b. daje wynik liczbowy równoważny operacji: `MOV $0,%eax`
- ☐ c. jest równoważne operacji `OR $0,%eax`
- ☐ d. jest równoważne operacji: `AND $0,%eax`
- ☒ e. daje wynik: `%eax=0` i flaga `Z=1`
- ☐ f. daje wynik zależny od wartości początkowej rejestru

Standardowe rozszerzenie plików z kodem źródłowym w assemblerze GAS to:

B i C

Wykonanie rozkazu `XOR %eax,%eax`:

B,D,E

Pytanie 3

Zakończone

Punkty: 5,00

🚩 Oflaguj pytanie

W których instrukcjach konieczny jest sufiks (przyrostek) określający wielkość/typ danych:

Wybierz jedną lub więcej:

- ☐ a. decl %edi
- ☐ b. cmpb \$5, x
- ☒ c. movq %eax, result
- ☒ d. movb \$10, %al
- ☐ e. incl counter

W których instrukcjach konieczny jest sufiks określający wielkość/typ danych:

Jeżeli pytanie nie odnosi się dokładnie do tych sufiksów, tylko mówi o jakichkolwiek oraz musi tylko zadziałać, nie mieć takiego wyniku jak wersja z sufiksem:

E

Pytanie 8

Zakończone

Punkty: 5,00

🚩 Oflaguj pytanie

Jedna linia pamięci cache we współczesnym procesorze x86-64 ma długość:

Wybierz jedną lub więcej:

- ☐ a. 512 bitów
- ☐ b. 64 bitów
- ☐ c. równą stronie pamięci wirtualnej
- ☒ d. długość linii cache zależy od długości rejestrów wektorowych, a tym samym od mikroarchitektury i generacji procesora
- ☐ e. 64 bajtów

Jedna linia pamięci cache we współczesnym procesorze:

E tak wikipedia mówi ale to d brzmi dobrze ktoś kto się na tym zna fajnie jakby odpowiedział na pytanie bo ja w sprzęt to słaby jestem sprawdziłem mój procesor i on ma 64 bajty.

1. Cechą architektury harwardzkiej jest:

- a) przechowywanie zmiennych w pamięci wspólnie z kodem programu,
- b) rozdzielanie pamięci danych od pamięci programu,**
- c) przechowywanie zmiennych w pamięci dynamicznej, a kodu programu w pamięci statycznej,
- d) posiadanie oddzielnych magistral: pamięci danych i pamięci algorytmu.**

2. Zapisać w systemie szesnastkowym liczbę binarną 1101010110111001

D5B9

3. Określić stan flag Sign (znak), Carry (przeniesienie), Zero i Overflow (nadmiar) po wykonaniu działania $(-7)+5$ (argumenty czterobitowe, kodowanie U2).

S=1 C=1 Z=0 O=0

4. Przesunięcie zawartości rejestru eax w lewo o dwa bity odpowiada:

- a) mnożeniu wartości w eax przez 2,
- b) mnożeniu przez 4 zawartości komórki pamięci wskazanej przez wartość w rejestrze,
- c) mnożeniu wartości w rejestrze przez 4,**
- d) dzieleniu wartości w rejestrze przez 4.

5. Cechą pamięci SDRAM jest:

- a) cykliczne odświeżanie zawartości,**
- b) podział na banki,**
- c) podział adresu na adres rzędu i kolumny,**
- d) przechowywanie zawartości po wyłączeniu zasilania,
- e) możliwość łatwej alokacji pamięci.**

6. Licznik rozkazów procesora (PC-program counter, IP-instruction pointer) jest to rejestr służący do:

- a) odmierzania czasu wykonywania danych procedury przez procesor,
- b) wyznaczania adresu skoku względnego,
- c) generowania przerwania co określoną liczbę wykonywanych instrukcji,
- d) przechowywania adresu kolejnej instrukcji do pobrania i wykonania.**

7. Stosując kod uzupełnień do dwóch (U2), w zmiennej ośmiobitowej można poprawnie zapisać wartość z zakresu:

- a) 0-255
- b) -128-127**
- c) -127-128

8. Liczbę zmiennoprzecinkową R można zapisać za pomocą liczb całkowitych: S-znak, M-mantysa-ułamek, W-wykładnik, P-podstawa systemu liczbowego powiązanych zależnościami:

- a) $R = S * M * P^W$**
- b) $R = P * M * S^W$
- c) $R = S * M * \log P(W)$

9. Jaki typ zależności w kodzie rozwiązuje technika przemianowania rejestrów (register renaming):

- a) WaR**
- b) RaW
- c) WaW**
- d) WaR i WaW**
- e) WaR i RaW
- f) wszystkie kombinacje powyższych

10. Korzyści odnoszone z obecności pamięci podręcznej cache w systemie komputerowym związane są z:

- a) lokalnością odniesień w kodzie programu, odpowiednią organizacją danych w pamięci RAM komputera,**
- b) lokalnością czasową – częstotliwością odwołań do określonych danych i kodu programu,**
- c) czasem dostępu do głównej pamięci operacyjnej,**
- d) mapowaniem portów I/O w przestrzeni adresowej procesora.

11. Cykl rozkazowy procesora składa się następującej sekwencji cykli maszynowych:

- a) pobranie instrukcji -> inkrementacja licznika rozkazów -> dekodowanie informacji -> wykonanie operacji -> zapis wyniku operacji i/lub ustawienie flag**
- b) pobranie instrukcji -> dekodowanie instrukcji -> ustawienie flag -> wykonanie operacji -> zapis wyniku
- c) pobranie instrukcji -> dekodowanie instrukcji -> wykonanie operacji -> zapis wyniku operacji i/lub ustawienie flag -> inkrementacja licznika rozkazów
- d) pobranie instrukcji -> dekodowanie instrukcji -> wykonanie operacji -> zapis wyniku operacji/ustawienie flag -> reset licznika rozkazów

12. Z punktu widzenia teorii układów cyfrowych procesor jest układem:

- a) sekwencyjnym,**
- b) proporcjonalnym,
- c) kombinacyjnym,
- d) liniowym,

13. Przerzutnik może być traktowany jako:

- a) podstawowy element funkcjonalny pamięci rejestru mogący przechowywać najmniejszą jednostkę informacji,**
- b) układ przełączający magistrale: danych i adresową,
- c) program obsługi kilku monitorów ekranowych w środowisku graficznym,
- d) element portu wyjściowego,**
- e) element portu wejściowego.**

14. Algorytm Write-Back:

- a) jest stosowany w obsłudze pamięci wirtualnej,
- b) nie jest stosowany w obsłudze pamięci cache,
- c) jest stosowany w kontrolerze przerwania,
- d) powoduje potencjalne problemy ze spójnością danych w pamięci cache i RAM,**

15. Obsługa zdarzenia typu błąd strony (page fault) odbywa się przez:

- a) zgłoszenie przerwania niemaskowalnego
- b) zgłoszenie wewnętrznego przerwania CPU- wyjątku,**
- c) polling
- d) użycie kanału DMA do przesłania brakującej strony.

16. Najnowsze procesory zgodne z x86 (i5, i7 itp)

- a) posiadają zintegrowany kontroler pamięci,**
- b) posiadają wiele jednostek wykonawczych o zróżnicowanej funkcjonalności,**
- c) mogą równolegle wykonywać kilkanaście instrukcji (w obrębie jednego rdzenia),
- d) posiadają pamięć cache L3 odrębną dla każdego rdzenia,
- e) posiadają zintegrowany mostek północny.**

1. W architekturze von Neumana:

- a) komórki pamięci mają adresy będące wyłącznie wielokrotnością podstawy użytego systemu liczbowego,
- b) bez analizy zawartości pamięci nie można jednoznacznie określić, czy dany blok pamięci zawiera kod programu, czy dane,**
- c) zmienne przechowywane są w komórkach pamięci o adresach parzystych, a kod programu – nieparzystych,
- d) zorganizowana jest pamięć operacyjna komputerów PC.**

2. Zapisać w systemie szesnastkowym liczbę binarną 1111101011001110
FACE

3. Określić stan flag Sign (znak), Carry (przeniesienie), Zero i Overflow (nadmiar) po wykonaniu działania $(-5)+7$ (argumenty czterobitowe, kodowanie U2).

S=0 C=0 Z=0 O=0

4. Pamięć podręczna cache L1:

- a) zoptymalizowana jest pod kątem minimalizacji czasu dostępu (CPU do danych w pamięci),**
- b) wykorzystuje pełną skojarzeniowość (odwzorowanie typu fully associative),
- c) wykorzystuje najczęściej algorytm write-through,
- d) jest współdzielona przez rdzenie procesora.

5. Wykonanie rozkazu xor eax, eax da w wyniku:

- a) eax=0 i flaga Z=1**
- b) eax=1 i flaga Z=0
- c) eax=0 i flaga Z=0
- d) wynik będzie zależny od wartości początkowej rejestru eax.

6. Procesory CISC cechuje:

- a) wiele różnych trybów adresowania,**
- b) prosta realizacja przetwarzania potokowego – dzięki dużej liczbie złożonych i specjalizowanych instrukcji,
- c) obecność instrukcji, upraszczających ręczne programowanie w języku assembler,**
- d) bardzo mały pobór prądu.

7. System przerwań zgodny z architekturą x86 obejmuje:

- a) przerwania chronione i wirtualne,
- b) przerwania niemaskowalne,**
- c) przerwania sprzętowe: zewnętrzne i wewnętrzne (wyjątki) oraz programowe,**
- d) przerwania modulowane fazą sygnały zegarowego,

8. Liczbę zmiennoprzecinkową R można zapisać za pomocą liczb całkowitych: S-znak, M-mantysa-ułamek, W-wykładnik, P-podstawa systemu liczbowego powiązanych zależnościami:

- a) $R = S * M * \log P(W)$
- b) $R = P * M * S^W$
- c) $R = S * M * P^W$**

9. Funkcje systemowe w Linuxie (system calls) uruchamia się wywołując przerwanie:

- a) programowe 80h**
- b) programowe 21h
- c) sprzętowe 80h
- d) sprzętowe 21h

10. Procesory ARM/Cores(?) stosowane obecnie na szeroką skalę np. w procesorach do urządzeń przenośnych mają architekturę typu:

- a) RISC**
- b) CISC
- c) SPARC
- d) CISC przekształconą wewnętrznie do RISC.

11. Jednostka arytmetyczno-logiczna (ALU) w procesorze służy do:

- a) wykonywania operacji m.in.: *, +, -, and, or, add, (?)**
- b) steruje przełączaniem bloków procesora między wykonywaniem instrukcji arytmetycznych i logicznych,
- c) może obliczać komórki pamięci i sterować wykonywaniem skoków warunkowych.**

12. DMA (Direct Memory Access) wprowadzono żeby:

- a) procesor nie musiał czynnie zajmować się transferem danych (I/O-pamięć),**
- b) maksymalizować przepustowość urządzenia I/O-pamięć,**
- c) poprawić efektywność działania systemu przerwań,
- d) poprawić efektywność działania pamięci cache,
- e) system efektywniej przełączał kontekst procesora.

13. Z przetwarzaniem potokowym wiąza się:

- a) hazardy danych,**
- b) konflikty w dostępie do różnych jednostek wykonawczych procesora,**
- c) utrata spójności pamięci cache,
- d) spekulacyjne wykonywanie instrukcji warunkowych/skoków,**
- e) uruchomienie procesów systemowych i użytkownika w dwóch oddzielnych potokach.

14. Problemem występującym w zapisie binarnym liczb wykorzystującym bit znaku (np. float, znak-moduł) jest występowanie:

- a) podwójnego zera (+0 i -0),**
- b) NaN (Not a Number),**
- c) symboli nieoznaczonych,**
- d) wyjątków zgłaszanych przez CPU.

15. Wykonując rozkaz mov eax,[var] (składnia Intel), do rejestru eax wczytana zostanie:

- a) wartość zmiennej var,**
- b) adres, pod którym umieszczona jest zmienna var,
- c) adres segmentu pamięci, w którym umieszczona jest zmienna var,
- d) wartość przechowywana w obszarze pamięci rozpoczynającym się od adresu określonego wartością var,

16. Cykl rozkazowy procesora składa się z następującej sekwencji cykli maszynowych:

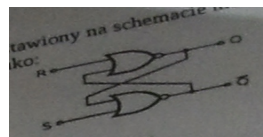
- a) pobranie instrukcji -> inkrementacja licznika rozkazów -> dekodowanie informacji -> wykonanie operacji -> zapis wyniku operacji i/lub ustawienie flag**
- b) pobranie instrukcji -> dekodowanie instrukcji -> ustawienie flag -> wykonanie operacji -> zapis wyniku
- c) pobranie instrukcji -> dekodowanie instrukcji -> wykonanie operacji -> zapis wyniku operacji i/lub ustawienie flag -> inkrementacja licznika rozkazów
- d) pobranie instrukcji -> dekodowanie instrukcji -> wykonanie operacji -> zapis wyniku operacji/ustawienie flag -> reset licznika rozkazów

- Wykonanie rozkazu „and eax0” daje wyniku:
 - Eax=0 i flaga Z=1**
 - Eax=1 i flaga Z=0
 - Eax bez zmian stan flagi Z będzie zależny od wartości rejestru eax
 - Eax=0 i flaga Z=0
 - Eax=1 i flaga Z=1
- Po operacji arytmetycznej (lub logicznej) wykonanie rozkazu skoku warunkowego „je etykieta” (jump if equal) możliwe tylko przy następującym ustawieniu flag:
 - Z=0 i S=0
 - Z=1 i S=1
 - Z=1 i S=0**
 - Z=0 i S=1
- Uszeregować interfejsy zaczynając od mającego największą przepustowość(1) do najwolniejszego
 - 3 USB 2.0
 - 1 PCI Express x16 3.0
 - 4 PS/2
 - 2 seria ata
- W architekturze x86 adresy portów I/O znajdują się w:
 - Przestrzeni adresowej pamięci operacyjnej
 - Przestrzeni adresowej urządzeń I/O**
 - Przestrzeni adresowej pamięci ROM
 - W tablicy deskryptorów portów
 - W dwóch, niezależnych przestrzeniach – osobno dla urządzeń I oraz O
- Podać liczbę wszystkich zależności danego typu, występujących w kodzie (składnia rozkazów MIPS: operacja, wynik_operacji, argument1, argument2)

1 add r1,r2,3	0 write after write
2 xor r2,r3,r1	3 write after read
3 and r3,r1,2	3 read after write
- FMA to:
 - Poprawione DMA (fast memory access)
 - Poprawione zarządzanie pamięcią (fast memory allocation)
 - Wykonywanie działań $a*b+c$ przy pomocy jednej instrukcji**
 - Zwiększenie dokładności poprzez redukcję liczby zaokrągleń pośrednich wyników w działaniu $a*b+c$ (fused multiply-add)**
- Komputer osobisty z wielordzeniowym procesorem (np.i7) może być uważany za:
 - System wieloprocessorowy o pamięci zozproszonej
 - System wieloprocessorowy o pamięci wspólnej**
 - Komputer, w którym dla każdego rdzenia dostępna jest cała przestrzeń adresowa**
 - Komputer z wieloma procesorami uniwersalnymi, dostosowanymi do wykonywania różnych zadań**
 - Komputer, w którym czas dostępu do pamięci operacyjnej zależy adresu konkretnej komórki i numeru rdzenia CPU
- Określić stan flag Sign(znak) Carry(przeniesienie)Zero i overflow(nadmiar) po wykonaniu działania $(-5)+3$ (argumenty czterobitowe, kodowanie U2)

S=1 C=1 Z=0 O=0

- Układ przestawiony na schemacie może znaleźć zastosowanie jako:



- Element rejestru**
 - Element porty wyjściowego**
 - Element portu wejściowego**
 - Element pamięci dynamicznej RAM
 - Element pamięci statycznej RAM**
- Bramka realizująca funkcję eXOR może znaleźć zastosowanie:
 - Przy generowaniu/sprawdzaniu bitu parzystości**
 - Jako „sterowana” zewnętrznym sygnałem bramka negacji**
 - W komparatorze (dekoderze równości argumentów)**
 - W generowaniu flagi overflow**
 - W generowaniu flagi Carry(przeniesienie)
 - W sprawdzeniu poprawności działania arytmetycznych w kodzie U2
 - Instrukcja przekształcająca wartość 01010101B w 8-bitowym rejestrze R na 10101010B jest:
 - Shr R,1
 - SarR,1
 - xorR,10101010B
 - xorR, 00000000B
 - xorR, 11111111B**
 - shlR,1**
 - Instrukcje push i pop (w CPU zgodnych z x86):
 - Związane są z obsługą stosu**
 - Związane są z transferem danych do/z pamięci**
 - Modyfikują wartość wskaźnika stosu**
 - Modyfikują wartość ramki stosu
 - Modyfikują znaczniki procesora
 - W procesorach Pentium4, Core2, i7 zastosowano wielowątkowość typu:
 - Fine-grained
 - Coarse-grained,
 - SMT(simultaneous MultiThreading)**
 - W komputerze z pamięcią cache działającą w oparciu a algorytm write-trhrough:
 - Gwarantowana jest maksymalna wydajność systemu
 - Każdy zapis z pamięci cache jest przenoszony do pamięci RAM w celu utrzymania spójności**
 - Dane z procesora do pamięci cache są przesyłane przez dodatkowy bufor
 - Praca pamięci cache jest zależna od konfiguracji pamięci wirtualnej
 - Cztery komórki pamięci o adresach od 00 do 03 zawierają kolejno bajty: 12 34 AA CD. Jaka liczba zapisana jest w pamięci, zakładając, że system pracuje w porządku big-endian?
 - CDAB3412
 - 1234ABCD**
 - 2143BADC
 - DCBA4321

1

Najważniejszą cechą charakteryzującą maszynę von Neumanna (w odróżnieniu od innych wczesnych modeli komputerów) jest:

- posiadanie rejestrów
- wykorzystanie licznika rozkazów
- potokowe przetwarzanie rozkazów
- przechowywanie rozkazów i danych w jednej pamięci

2

Pamięć o dostępie swobodnym definiujemy jako pamięć:

- dostępna dla każdego programu
- o niezależnym bezpośrednim dostępie do każdej komórki
- umożliwiającą dostęp do komórek każdemu procesorowi maszyny
- będącą teoretycznym modelem nieskończonej taśmy z maszyny Turinga

3

Procesor chcąc uzyskać dostęp do komórki pamięci RAM może posłużyć się:

- symboliczną nazwą komórki
- adresem komórki zapisanym w postaci bitowej
- adresem komórki przechowywanym w rejestrach
- specjalnym rejestrem przechowującym adres w postaci symbolicznej

4

ALU (jednostka arytmetyczno-logiczna) służy do:

- dekodowania i wykonywania rozkazów
- wykonywania wszystkich rozkazów
- wykonywania m.in. operacji dodawania, mnożenia itp.
- wykonywania m.in. operacji skoku
- wykonywania m.in. operacji pobierania z pamięci RAM

5

Elementem jednostki centralnej w maszynie von Neumanna **nie jest**:

- jednostka sterująca
- zegar
- zestaw rejestrów
- pamięć podręczna
- wewnętrzna magistrala
- jednostka arytmetyczno-logiczna

6

Licznik rozkazów:

- jest to układ liczący wykonane rozkazy
- jest to rejestr zawierający adres kolejnego rozkazu do wykonania
- jest to układ służący do przewidywania liczby rozkazów „omijanych” przy rozkazie skoku
- służy m.in. do realizacji rozkazu skoku

7

Akumulator (w procesorze) jest to:

- rejestr przechowujący adresy rozkazów do wykonania
- rejestr sumujący wyniki poprzednich operacji
- rejestr przechowujący wyniki dodawania
- rejestr przechowujący dane i wyniki operacji ALU
- rejestr będący domyślnym argumentem rozkazów ALU

8

Rozkaz pobierany z komórki pamięci przez procesor i następnie wykonywany:

- zawiera nazwę (numer) rozkazu i zawsze wskazania adresów argumentów
- zawiera nazwę (numer) rozkazu i ewentualnie wskazania adresów argumentów (jeśli są rozkazowi potrzebne)
- zawiera nazwę (numer) rozkazu i ewentualnie nazwę (numer) rozkazu zawierającego adresy argumentów
- zawiera tylko nazwę rozkazu, adresy argumentów są w kolejnych komórkach pamięci

9

Wykonanie każdego rozkazu składa się etapów, wśród których zawsze występują:

- pobranie rozkazu
- zdekodowanie rozkazu
- pobranie argumentów z pamięci
- wykonanie rozkazu
- zapis wyniku do pamięci

10

Podstawowymi operacjami, koniecznymi do realizacji algorytmów, występującymi na listach rozkazów wszystkich procesorów są:

- operacje arytmetyczne (dodawanie, mnożenie itp.)
- operacje logiczne (konjunkcja, alternatywa, itp.)
- operacje na znakach (odczyt, zapis, zmiana kodowania, itp.)
- operacje na obrazach (odczyt, wyświetlanie, zmiana parametrów itp.)
- operacje na plikach (otwieranie, zamykanie, odczyt, zapis itp.)
- operacje na komórkach pamięci RAM (odczyt zawartości, zapis, itp.)
- operacje skoku

11

Rozkaz skoku jest realizowany przez procesor poprzez:

- przesunięcie kodu w pamięci operacyjnej
- podmianę aktualnie wykonywanego rozkazu na rozkaz będący celem skoku
- wykonanie procedury systemowej realokacji odniesień do pamięci
- zmianę zawartości rejestru licznika rozkazów, zawierającego adres kolejnego rozkazu do wykonania
- przełączenie kontekstu: zmianę zawartości rejestrów i struktur danych obsługujących aktualnie wykonywany fragment kodu

12

Maszyna Turinga jest abstrakcyjnym modelem maszyny obliczeniowej, o takich samych możliwościach jak współczesne komputery; charakteryzuje się ją jako maszynę (zaznacz każde właściwe określenie):

- czytającą program z nieskończonej taśmy
- czytającą dane z nieskończonej taśmy
- której działanie polega na zmianach jej stanu
- o skończonej liczbie stanów

1

Bramka w informatyce to:

- układ elektroniczny realizujący podstawowe jednobitowe operacje logiczne
- układ elektroniczny realizujący podstawowe wielobitowe operacje logiczne
- układ elektroniczny realizujący podstawowe wielobitowe operacje arytmetyczne
- układ elektroniczny służący do zachowania i zmiany stanu jednego ze swoich wyjść poprzez odpowiednie sygnały wejściowe
- układ do przechowania i zmiany pojedynczego bitu

2

Przerzutnik w informatyce to:

- układ elektroniczny realizujący podstawowe jednobitowe operacje logiczne
- układ elektroniczny realizujący podstawowe wielobitowe operacje logiczne
- układ elektroniczny realizujący podstawowe wielobitowe operacje arytmetyczne
- układ elektroniczny służący do zachowania i zmiany stanu jednego ze swoich wyjść poprzez odpowiednie sygnały wejściowe
- układ do przechowania i zmiany pojedynczego bitu

3

Funkcja logiczna „p XOR q”, dla kolejnych par wartości p i q równych (0,0), (0,1), (1,0), (1,1) przybiera wartości:

- 0,0,0,0 0,1,1,0 1,0,0,1 1,0,0,0 0,0,0,1 1,1,1,0 0,1,1,1

4

Funkcja logiczna „p NAND q”, dla kolejnych par wartości p i q równych (0,0), (0,1), (1,0), (1,1) przybiera wartości:

- 0,0,0,0 0,1,1,0 1,0,0,1 1,0,0,0 0,0,0,1 1,1,1,0 0,1,1,1

5

ALU jest to:

- układ elektroniczny realizujący podstawowe jednobitowe operacje logiczne
- układ elektroniczny realizujący podstawowe wielobitowe operacje logiczne
- układ elektroniczny realizujący podstawowe wielobitowe operacje arytmetyczne
- układ elektroniczny służący do zachowania i zmiany stanu jednego ze swoich wyjść poprzez odpowiednie sygnały wejściowe
- układ do przechowania i zmiany pojedynczego bitu
- żadna odpowiedź nie jest prawidłowa

6

Rejestr jest to:

- układ elektroniczny do przechowywania spisu rozkazów procesora
- układ elektroniczny do przechowywania danych złożony z przerzutników
- podstawowa komórka pamięci wewnętrznej procesora
- układ do przechowywania stanu pamięci podręcznej procesora
- żadna odpowiedź nie jest prawidłowa

7

Zgodnie z prawem Moore'a liczba tranzystorów umieszczanych w pojedynczym układzie scalonym podwaja się co x miesięcy, gdzie x wynosi:

- 7 8 17 18 27 28 47 48

8

Prawo Moore'a stwierdza, że co stałą liczbę (ok. 18) miesięcy podwaja się:

- liczba tranzystorów umieszczanych w pojedynczym układzie scalonym
- wydajność procesorów
- częstotliwość taktowania zegara procesora
- rozmiar pamięci podręcznej procesora
- pojemność twardego dysku

9

Wskaż prawidłową kolejność etapów przetwarzania potokowego (ID – dekodowanie rozkazu, IE – wykonanie rozkazu, IF – pobranie rozkazu, OF – pobranie argumentów, WB – zapis wyniku):

- IE ID OF IF WB
- ID IF WB IE OF
- IE IF WB ID OF
- WB IE IF ID OF
- IF ID OF IE WB

10

Potokowe przetwarzanie rozkazów oznacza przetwarzanie:

- potoku rozkazów – kolejny rozkaz po zakończeniu poprzedniego
- zbliżone do pracy na taśmie produkcyjnej – pojedyncza jednostka funkcjonalna procesora realizuje tylko część przetwarzania rozkazu
- dzięki któremu procesor może współbieżnie przetwarzać wiele rozkazów
- wymagające istnienia złożonych procesorów o wielu jednostkach funkcjonalnych
- takie jak w kartach graficznych (inaczej przetwarzanie strumieniowe)

11

Superskalarność procesora oznacza:

- bardzo rozbudowane potoki przetwarzania (kilkanaście etapów)
- istnienie kilku potoków przetwarzania
- posiadanie więcej niż jednej jednostki funkcjonalnej ALU
- możliwość wykonywania instrukcji poza kolejnością
- wsparcie sprzętowe wielowątkowości

12

Możliwości równoległości na poziomie wykonania pojedynczego rozkazu (*Instruction Level Parallelism*) współczesnych procesorów (rdzeni) są ograniczone do następującej liczby rozkazów kończonych w każdym cyklu zegara:

- 1 kilka kilkanaście kilkadziesiąt kilkaset

13

Producenci procesorów ogólnego przeznaczenia przestali zwiększać częstotliwość pracy procesorów z powodu:

- zbyt wielu etapów w potokowym przetwarzaniu rozkazów, utrudniających zrównoleglenie kodu
- zbyt wysokiego poziomu wydzielania ciepła
- barier technologicznych w taktowaniu układów elektronicznych
- niemożności zagwarantowania odpowiednio szybkiej pracy pamięci podręcznej
- żadna z wymienionych możliwości

Płyta kuchenki elektrycznej produkuje ok. 10 W/cm² natomiast współczesne procesory około (w W/cm²):

- 1 10 100 1000 10000

14

Procesory wielordzeniowe:

- są nazywane inaczej układami scalonymi wieloprocessorowymi
- pracują w modelu SIMD (*single instruction multiple data*)
- nigdy nie przekroczą liczby rdzeni ok. kilkunastu
- nie posiadają pamięci podręcznej L2 i L3
- żadne stwierdzenie nie jest prawdziwe

15

Rdzeń procesora wielordzeniowego:

- jest bardzo zbliżony do dawnych procesorów (jednordzeniowych), ale nie potrafi wykonywać bardziej złożonych rozkazów
- jest bardzo zbliżony do dawnych procesorów (jednordzeniowych), ale bez pamięci podręcznej L2 i L3
- wydziela znacznie mniej ciepła niż procesor jednordzeniowy o tej samej częstotliwości pracy
- może mieć własny dostęp do fragmentu pamięci głównej (architektura NUMA)
- żadne stwierdzenie nie jest prawdziwe

1

W celu realizacji programów procesory zapisują:

- kod i dane w systemie dwójkowym
- kod w systemie szesnastkowym, dane w systemie dwójkowym
- kod w systemie dwójkowym, dane w systemie szesnastkowym
- kod i dane w systemie szesnastkowym

2

System szesnastkowy zapisu liczb:

- jest sposobem przechowywania liczb przez komputer
- jest używany do zapisu liczb binarnych w formie czytelniejszej dla ludzi
- ma te same cyfry co układ dziesiętny
- ma cyfry: 0, 1, 2, ..., 9, a, b, c, d
- jest wygodny dla programistów, ponieważ jednej cyfrze układu szesnastkowego odpowiadają cztery cyfry układu dwójkowego
- jest układem pozycyjnym, gdzie kolejnym pozycjom (cyfrom) w liczbie przyporządkowane są kolejne potęgi 16 (1, 16, 256, itd.)

3

Informacja, że procesor jest k-bitowy (np. 32-bitowy) oznacza m.in. że:

- posiada k-bitowe rejestry
- może adresować pamięć od 0 do 2^k do potęgi (k-1)
- liczby zmiennoprzecinkowe są przechowywane za pomocą k-bitów
- adresy są przechowywane za pomocą k-bitów
- znaki są przechowywane za pomocą k-bitów

4

Znaki zapisywane są poprzez podanie związanej z nimi liczby czyli kodu znaku. Liczba znaków jest równa 2^k do potęgi k, gdzie k jest liczbą bitów użytych do kodowania znaku. W systemach ASCII i UTF liczby znaków to:

- 64 i 65536
- 2^7 i 65536
- 256 i 2^{16}
- 2^7 i 2^{16}
- 256 i 65536

5

Największą liczbę zapisaną w postaci 32-bitów uzyskuje się przez zastosowanie kodowania jak dla:

- liczb naturalnych
- liczb całkowitych w systemie ze znakiem
- liczb całkowitych w systemie uzupełnień do dwóch
- liczb zmiennoprzecinkowych pojedynczej precyzji
- liczb zmiennoprzecinkowych podwójnej precyzji

6

Sposób kodowania liczb zmiennoprzecinkowych za pomocą skończonej liczby bitów (w praktyce 32 lub 64) oznacza, że m.in.:

- można zapisać tylko skończoną ilość liczb podczas gdy liczb rzeczywistych jest nieskończenie wiele
- na osi liczbowej pomiędzy liczbami zmiennoprzecinkowymi są odstępy (między liczbami rzeczywistymi nie ma, tzn. nie wszystkie liczby rzeczywiste dają się zapisać jako zmiennoprzecinkowe)
- nie wszystkie operacje wykonalne na liczbach rzeczywistych mogą być wykonywane na liczbach zmiennoprzecinkowych
- wynik operacji na liczbach zmiennoprzecinkowych może być liczbą nie dającą się zapisać w postaci zmiennoprzecinkowej
- wyniki operacji na liczbach zmiennoprzecinkowych po zapisaniu w postaci zmiennoprzecinkowej mogą być niedokładne

7

Zapis liczby zmiennoprzecinkowej x, polega na zapisie dwóch liczb całkowitych, obliczonych z wykorzystaniem wzoru:

- $x = a/b^2$
- $x = a \cdot 2^b$
- $x = a \cdot 10^b$
- $x = a \cdot \log_2(b)$
- $x = a \cdot \log_{10}(b)$

8

Big-endian i little-endian to nazwy dwóch różnych sposobów zapisu:

- liczb naturalnych i całkowitych ze względu na liczbę bitów użytych do zapisu
- liczb naturalnych i całkowitych ze względu na kolejność umieszczania bitów w komórkach pamięci
- liczb całkowitych ze względu na sposób uwzględnienia liczb mniejszych od zera
- liczb całkowitych ze względu na liczbę bitów użytych do zapisu

9

Liczby zmiennoprzecinkowe pojedynczej (32 bity) i podwójnej (64 bity) precyzji różnią się:

- sposobem uwzględnienia znaku liczby
- kolejnością zapisywanych bajtów w komórkach pamięci
- rodzajem rejestrów i jednostek ALU stosowanych do ich obsługi przez procesor
- liczbą bitów użytych do zapisu wykładnika potęgi dwójki
- zakresem czyli największymi i najmniejszymi możliwymi do zapisania liczbami dodatnimi i ujemnymi
- ilością bitów użytych do zapisu liczby przez którą mnożona jest potęga dwójki
- dokładnością czyli odległością dwóch kolejnych liczb

10

Ze względu na różne sposoby zapisu liczb różnych rodzajów (naturalnych, ułamkowych itp.), a także liczb tego samego rodzaju:

- procesory używają specjalnych adresów dla każdego sposobu zapisu liczb
- w programach wprowadza się pojęcie typu zmiennej i stosuje się zmienne różnych typów
- rozróżnia się systemy 32 i 64 bitowe
- systemy mogą różnić się szczegółami zapisu liczb, nawet liczb tego samego rodzaju (np. naturalnych)
- przy przesyłaniu liczby z jednego systemu do drugiego, każdy z nich może ją odczytać w inny sposób

11

W programach komputerowych występują zmienne, czyli symbole które mogą przybierać różne wartości. Analizując wykonanie kodu przez procesor można przyjąć, że:

- zmiennej odpowiada pewien obszar pamięci operacyjnej
- nazwa zmiennej nie ma znaczenia dla procesora
- typ zmiennej nie ma znaczenia dla procesora
- procesor odwołuje się do zmiennej tylko poprzez jej adres w pamięci
- rozmiar zmiennej w pamięci dla danego systemu (np. 32-bitowego) jest taki sam dla wszystkich typów

12

W przypadku wykonania w programie operacji arytmetycznej niedozwolonej lub operacji o wyniku, którego nie da się zapisać w sposób standardowy (np. dzielenie przez zero lub liczbę bardzo bliską zero) zgodnie ze standardami i konwencją system może zgłosić błąd oraz:

- kontynuować działanie programu bez zmian
- kontynuować działanie programu wstawiając specjalne symbole w miejsce wyników
- dokonać korekty programu tak żeby błąd nie wystąpił
- przerwać działanie programu
- przerwać pracę procesora (czyli systemu)

Różne zadania w stylu: liczba w notacji zapisana jako ma w notacji zapis:

(używane notacje dla liczb naturalnych: dwójkowa, szesnastkowa, dziesiętna; dla liczb zmiennoprzecinkowych notacja naukowa z podstawą 10) np.

a

Liczba w notacji dwójkowej zapisana jako 11101100 w notacji dziesiętnej to:

- 194
- 236
- 217
- 115
- 214
- 222

d

Liczba 222 w notacji dwójkowej to:

- 10110110
- 11011110
- 10010111
- 11100101
- 10101101
- 11001010

e

Liczba 1234,5678 w notacji naukowej znormalizowanej, tzn. z jedną cyfrą przed przecinkiem, ma postać:

- $1,2345678 \cdot 10^0$
- $1,2345678 \cdot 10^3$
- $1,2345678 \cdot 10^{-3}$
- $1234567,8 \cdot 10^{-3}$
- $1234567,8 \cdot 10^3$

Ile bitów potrzeba, żeby dokładnie zapisać liczbę?

-

1

Język asemblera:

- jest językiem programowania tzw. niskiego poziomu
- jest sposobem zapisu w postaci symbolicznej (łatwiejszej do zrozumienia dla programisty) rozkazów maszynowych procesora
- nie posiada rozkazów skoku (zastąpione przez pętle)
- posługuje się adresami komórek pamięci i nazwami rejestrów zamiast nazw zmiennych
- wprowadza własne nazwy zmiennych zrozumiałe dla procesora

2

W programach zapisanych w językach programowania wyższego poziomu programiści posługują się zmiennymi, czyli symbolami, które mogą przyjmować różne wartości (podobnie jak w matematyce). W języku asemblera (czyli w rozkazach, które rzeczywiście wykonuje procesor) zamiast tego używa się odniesień do komórek pamięci (wewnętrznej procesora – rejestrów i zewnętrznej RAM) a dostęp do zawartości komórki uzyskuje się podając:

- adres wewnętrzny rejestru w zapisie binarnym, np. 1101001
- nazwę rejestru np. `%eax`
- adres komórki pamięci w zapisie szesnastkowym np. `$567af3b2`
- adres komórki pamięci przechowywany w rejestrze, stosując notację nawiasów oraz podając adres wewnętrzny rejestru, np. (1101001)
- adres komórki pamięci przechowywany w rejestrze, stosując notację nawiasów oraz podając nazwę rejestru np. (`%ebx`)

3

Rozkazy języka asemblera, przedstawiające w postaci symbolicznej rozkazy wykonywane przez procesor, zawierają nazwę rozkazu, określającą wykonywaną operację oraz argumenty operacji (jeśli występują), którymi mogą być stałe (liczby, znaki, napisy), zawartości rejestrów, zawartości komórek pamięci oraz:

- etykiety rozkazów,
- adresy rozkazów
- nazwy procedur
- adresy procedur
- nazwy urządzeń wejścia/wyjścia

4

Zapis (`%ecx`) w języku asemblera (notacja AT&T) oznacza:

- wartość zmiennej `ecx`
- adres procedury o nazwie `ecx`
- zawartość rejestru `ecx`
- zawartość komórki pamięci o adresie w rejestrze `ecx`
- adres obiektu `ecx`

5

Podstawowymi rozkazami wykonywanymi przez procesor są m.in. rozkazy:

- skoku (bezw warunkowego - `jmp` i warunkowego – `jeq`, `jl`, `jge`, itp.)
- przeniesienia danych do lub z rejestrów (`mov`, `load`, `store`, itp.)
- operacje logiczne (`and`, `or`, `xor`, itp.)
- operacji arytmetycznych (`add`, `mul`, `sub`, itp.)
- operacje graficzne (`draw`, `clear`, itp.)
- obsługi wywołania procedur (`call`, `ret`, itp.)

Architektura hardwardzka:

- Rozdzielanie pamięci danych od pamięci programu (2 oddzielne szyny)
- Przechowywanie zmiennych w pamięci dynamicznej, a kodu programu w pamięci statycznej (bo 2 oddzielne szyny)
- Posiadanie oddzielnych magistral: pamięci danych i pamięci algorytmu
- Prostsza budowa (w stosunku do von Neumanna)
- Większa szybkość.
- Często wykorzystywana przy dostępie procesora do pamięci cache.

Architektura von Neumana:

- utrzymywanie w pamięci komputera zarówno programu, jak i danych, które są przechowywane
- w kodzie dwójkowym, a ich przetwarzanie odbywa się w arytmometrze.

Według tej koncepcji komputer składa się z 3 podstawowych części:

- procesor – z wydzieloną częścią sterującą oraz częścią arytmetyczno-logiczną (ALU)
- pamięć – dane i instrukcje są przechowywane we wspólnej pamięci w postaci binarnej
- urządzenia wejścia/wyjścia

ALU - jednostka arytmetyczno -logiczna układ cyfrowy będący jedną z głównych części procesora prowadząca proste operacje arytmetyczne (dodawanie, odejmowanie) oraz operacje logiczne (AND, OR, NOT, XOR). Zazwyczaj ALU ma dwa wejścia dla pary argumentów i jedno wyjście dla wyniku.

CPU - (Central Processing Unit) - urządzenie cyfrowe sekwencyjne potrafiące pobierać dane z pamięci, interpretować je i wykonywać jako rozkazy

- Bez analizy zawartości pamięci nie można jednoznacznie określić, czy dany blok pamięci zawiera kod programu czy nie (wynika to z budowy maszyny. W pamięci dane i instrukcje są wspólnie przechowywane)
- Zorganizowana jest pamięć operacyjna komputerów PC (praktycznie wszystkie komputery są budowane według pomysłu von Neumanna)

Zamiana BIN <-> HEX

BIN -> HEX

* dzielisz (cała liczba)

29	1
14	0
7	1
3	1
1	1
0	1

BIN -> DEC

* rozpisz
* dodaj

BIN -> HEX (Algo리즘)

* zapisz

0110	1110	1010
------	------	------

* podziel

* mierze dodaj zero

* wypisz **8421**

* zamień litery

np 10 = A
14 = E itp.
6 = 6

HEX -> BIN

* rozpisz

B4F

8421 8421 8421

* po dodaniu tych cyfry suma

1011 0100 1111

BIN -> OCT (8)

* rozpisz tylko

421

OCT -> BIN

247

417 417 417

010 100 111

Flagi:

- S (sign flag)** – flaga znaku, wynik pozytywny gdy ujemny. (dla wyniku -1 przyjmie wartość 1)
- C (carry flag)** – flaga przeniesienia, **wynik operacji zawiera się w większej niż dostępna liczbie bitów.** (np. dla 4bitowego wyrażenia, gdzie zakres wynosi 0..15 wyjdzie nam 16, to **C=1**) **NIE INTERPRETUJE ZNAKU, MA WYWALONE NA MINUSY!!**
- Z (zero flag)** – flaga zera, **pozytywne jeśli wynik = 0**
- O (overflow)** – flaga przepełnienia, podobna do flagi przeniesienia, jednak **odnosi się do operacji ze znakiem.** (zakres <-8,7> dla 4bitowego słowa)

Przesunięcie zawartości rejestru eax w lewo o 2 bity spowoduje:

mnożeniu wartości w rejestrze przez 4

Przesuwamy dwa w lewo, więc mamy
(0001 = 0100) - "I pyk z jedyinki zrobiła nam się czwóreczka" ~Maciej

Cechy pamięci SDRAM (rodzaj pamięci DRAM)

- Pracuje **synchronicznie** z magistralą systemową, w przeciwstwie do innych DRAM, które **asynchronicznie**
- Cykliczne odświeżanie zawartości
- Podział na banki
- Podział adresu na **adres rzędu** i **adres kolumny**
- Możliwość **łatwej alokacji pamięci**

Licznik rozkazów procesora (PC-program counter = IP-instruction pointer)

- rejestr, przechowuje informacje o tym, **w którym obecnie miejscu** sekwencji znajduje się procesor.
- **Przechowuje on adres kolejnej instrukcji** (w starszych modelach procesorów terażniejszej).
- Przez modyfikacje tego rejestru implementuje się skoki, skoki warunkowe, pętle i podprogramy.

Standard U2

W standardzie U2 **najstarszy bit** świadczy o znaku liczby. (1=liczba ujemna, 0=liczba dodatnia)
Jako, że na miejscu najstarszego bitu mamy znak, to **zakres naszej liczby to <-128,127>**.
1000 0000=-128 (najstarsza jedynka traktowana i jako znak, i jako liczba (-2^7). Nie jest to zero, ponieważ 0= 0000 0000, a w systemie tym występuje tylko jedno zero)
1000 0001(U2) = -127 (-128+1)
0111 1111(U2) =127 (1+2+4+8+16+32+64)

Zapis liczby zmiennoprzecinkowej, za pomocą ca łkowitych

R =S*M*P^W

S-znak M-znormalizowana mantysa, ułamek P-podstawa systemu liczbowego W-wykładnik

Technika przemianowania rejestrów (register renaming)

- używana w celu uniknięcia niepotrzebnego szeregowego wykonania instrukcji narzuconego przez wykorzystanie tych samych rejestrów procesora przez następujące po sobie instrukcje.
- Powodem powstawania hazardów jest wielokrotne używanie tych samych rejestrów do przechowywania różnych wartości

Dostępne hazardy:

- RAW** - read after write - NIE USUWA
- WAR** - write after read - OK
- WAW** - write after write - OK

Rozwiązuje to poniższe: (usuwa te hazardy)

- A)WaR** - nie musimy zapisywać nic w tym rejestrze po odczycie, bo specjalnie po to użyliśmy kolejnego rejestru
- c)WaW** - skoro dodaliśmy nowy rejestr na przechowywanie danych, to zapisany wcześniej rejestr nie jest nadpisany kolejnym
- d)War i WaW**

Pamięć cache: korzyści

najważniejsze:

- **zmniejszenie czasu dostępu**
- **wbudowany algorytm inteligentnego przechowywania danych = lepsze wykorzystanie**
- **lokalność odniesień w kodzie programu**

inne:

- Odpowiednna organizacja danych w pamięci RAM komputera
- Lokalność czasowa -częstotliwość odwołań do określonych danych i kodu programu
- Czas dostępu do głównej pamięci operacyjnej

Cykl rozkazowy procesora: sekwencja cykli maszynowych

1. Pobranie rozkazu

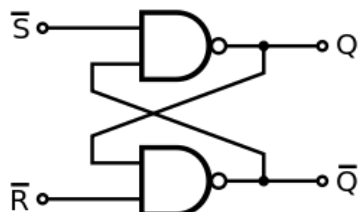
2. (Inkrementacja licznika rozkazów)
3. Dekodowanie rozkazu
4. Wykonanie rozkazu
5. Zapis wyniku i/lub ustawienie flag

Procesor jest układem sekwencyjnym

Jak najbardziej

Przerzutnik może być traktowany jako

- Podstawowy element funkcjonalny pamięci rejestru: może przechowywać **najmniejszą jednostkę informacji**
- **Element portu wejściowego / wyjściowego**
- małe ilości danych, do których musi być zapewniony ciągły dostęp.



Może to być element rejestru, element portu w/w, element pamięci statycznej RAM (**nie może być dynamicznej RAM**)

Wyjątek "Page Fault", czyli tzw. błąd stronicowania,

jest zgłaszany przez procesor wtedy, kiedy nastąpi nieprawidłowe odwołanie do jakiegoś liniowego adresu pamięci. Nieprawidłowe oznacza, że strona wyznaczana przez dany adres nie jest obecna.

Przerwanie lub żądanie przerwania to sygnał powodujący zmianę przepływu sterowania, niezależnie od aktualnie wykonywanego programu. Pojawienie się przerwania powoduje wstrzymanie aktualnie wykonywanego programu i wykonanie przez procesor kodu procedury obsługi przerwania.

-Tak więc ze względu na wystąpienie błędu polegającego na nieprawidłowym odwołaniu do adresu pamięci, do procesora wysyłany jest sygnał żądania przerwania. (ODP: zgłoszenie wewnętrznego przerwania CPU-wyjątku)

Najnowsze procki zgodne z x86 (i5, i7 itp.)

- Posiadają zintegrowany kontroler pamięci
- Posiadają wiele jednostek wykonawczych o zróżnicowanej funkcjonalności
- Posiadają zintegrowany mostek północny
- **NIE POSIADAJĄ** pamięci cache L3 oddzielnej dla każdego rdzenia
- **NIE MOGĄ** równolegle wykonywać kilkunastu instrukcji na jednym rdzeniu

Procesory CISC:

- Wiele różnych trybów adresowania
- Obecność instrukcji, które **UPRASZCZAJĄ** ręczne programowanie w assemblerku <3

System przerwania x86 obejmuje:

- Przerwanie **niemaskowalne**
- Przerwanie **sprzętowe**: zewnętrzne i wewnętrzne(wyjątki) oraz **programowe**
- **NIE OBEJMUJE** przerwania chronionych, wirtualnych, modulowanych fazą sygnału zegarowego

Przerwanie aby uruchomić system calls w Linuxie to przerwanie:

Programowe 80h

Procki ARM/Cores stosowane obecnie na szeroką skalę np. w prockach do urządzeń przenośnych mają architekturę typu RISC - najbardziej wydajne = najbardziej popularne

ALU - jednostka arytmetyczno -logiczna układ cyfrowy będący jedną z głównych części procesora

- proste operacje arytmetyczne (dodawanie, odejmowanie)
- operacje logiczne (AND, OR, NOT, XOR).
- sterowaniem skoków warunkowych.

Zazwyczaj ALU ma dwa wejścia dla pary argumentów i jedno wyjście dla wyniku.

DMA DIRECT MEMORY ACCESS

- ma za zadanie **odciążyć procesor główny od przesyłania danych** (np. z urządzenia wejściowego do pamięci). Procesor może w tym czasie zająć się innymi działaniami, wykonując kod programu pobrany uprzednio z pamięci RAM do pamięci podręcznej.
- Specjalizowane układy wspomagające DMA (np. te spotykane w PC), potrafią kopiować obszary pamięci dużo szybciej niż uczyniłby to programowo procesor główny.

ODP:wprowadzono to po to żbby

- a) **procesor nie musiał czynnie zajmować się transferem danych (I/O-pamięć)**
- B) **maksymalizować przepustowość urządzenia I/O-pamięć**

Przetwarzanie potokowe:

sposób przetwarzania rozkazów w procesorach, w którym do przetwarzania zastosowano potok podzielony na etapy. W przeciwieństwie do klasycznego (sekwencyjnego) sposobu przetwarzania rozkazów gdzie każdy jest pobierany, dekodowany, wykonywany oddzielnie, w przetwarzaniu potokowym w procesorze następuje **jednoczesne przetwarzanie kilku rozkazów jednocześnie**. Każdy z rozkazów znajduje się w danym cyklu maszynowym w innym etapie przetwarzania i dzięki temu wyniki operacji mogą być produkowane niemal w każdym cyklu.

Wiążą się z nim:

- **hazardy danych**
- **konflikty w dostępie do różnych jednostek wykonawczych procesora**
- **spekulacyjne wykonywanie instrukcji warunkowych/skoków**

Zapis binarny wykorzystujący bit znaku np float, znak-moduł - problemy:

- podwójne zero +0 -0
- NaN (not a number)
- symbol nieoznaczony
- **NIE MA PROBLEMU z wyjątkami od cpu**

```
-----
cmp eax,ebx
je wynik
```

je musi mieć flagę Z=1 (oraz S=0) bo różnica ta (wynik cmp) musi byc równa zero

Uszeregować wg największej przustowości:

1. PCI EXPRESS x16 3.0
2. Seria ata
3. USB 2.0
4. PS/2

W architekturze x86 adresy portów I/O

znajdują się w przestrzeni adresowej urządzeń I/O

FMA (alternatywna nazwa: FMAD)

to *zabezpieczona* (ang. *fused*) operacja typu **pomnóż-i-dodaj wykonywana w jednym kroku**, podczas którego wykonuje się tylko jedno zaokrąglenie. Oznacza to, że w instrukcji $a = b + c * d$ operacje mnożenia i dodawania wykonywane są dokładnie (bez żadnych zaokrągleń) i dopiero wynik dodawania zaokrąglany jest tak, by mógł być zapisany w a.

FMAD **przyspiesza i poprawia dokładność** wielu algorytmów numerycznych, w których wyznacza się sumy iloczynów:

Zwiększenie dokładności następuje poprzez **redukcję liczby zaokrągleń pośrednich wyników** w działaniu $a*b+c$ (**tylko raz zaokrągla**)!!!!

Komputer z prockiem wielordzyniowym: (np. i7)

- To system wielprocesorowy o **pamięci wspólnej (NIE rozproszonej!!)**
- Dla każdego rdzyna dostępna jest cała przestrzeń adresowa
- Komputer z wieloma procesorami uniwersalnymi, dostosowanymi do wykonywania różnych zadań
- czas dostępu do pamięci operacyjnej **NIE ZALEŻY** od adresu konkretnej komórki i numeru rdzyna CPU

Instrukcje push i pop

- związane są z obsługą stosu
- związane są z transferem danych do i z pamięci
- modyfikują wartość wskaźnika stosu
- **NIE MODYFIKUJĄ ramki stosu**
- **NIE MODYFIKUJĄ znaczników procesora**

Instrukcje Logical shift

Shift left i shift right

shlR, 1 (przesuwa wartosc rejestru R o 1 w lewo)

shrR, 1 (j/w ale w prawo)

Inne pytania:

Bramka realizująca funkcję eXOR może znaleźć

zastosowanie:

- a) Przy generowaniu/sprawdzaniu bitu parzystości
- b) Jako „sterowana” zewnętrznym sygnałem bramka negacji
- c) W komparatorze (dekoderze równości argumentów)
- d) W generowaniu flagi overflow
- e) W generowaniu flagi Carry(przeniesienie) **NIE**
- f) W sprawdzeniu poprawności działania arytmetycznych w kodzie U2 **NIE**

W procesorach Pentium4, Core2, i7 zastosowano

wielowątkowość typu:

- a) Fine-grained **NIE**
- b) Coarse-grained, **NIE**
- c) SMT(simultaneous MultiThreading)

Algorytmy zapisu:

Algorytm Write-Through: (zapis jednoczesny)

- Wszystkie operacje zapisu są rprowadzone jednocześnie do pamięci głównej i podręcznej
- Dzięki temu dane są zawsze aktualne w pamięci głównej
- Jakikolwiek inny moduł [procesor - pamięć podręczna] może monitorować przesyłanie do pamięci głównej w celu zchowania spójności pamięci podręcznej
- WADA: generuje znaczny przepływ danych do pamięci i może powodować wystąpienie wąskich gardeł

Algorytm Write-Back: (zapis opóźniony)

- minimalizuje ilość operacji zapisu do pamięci.
- aktualizuje się tylko pamięć podręczną.
- dane do pamięci głównej zapisuje się dopiero wtedy, gdy nie wystarcza już miejsca w pamięci podręcznej na ich dalsze magazynowanie.
- problem jest z tym że zawsze część pamięci głównej jest nieaktualna
- przez to dostęp przez I/O jest możliwy tylko za pomocą pamięci podręcznej

WAŻNE: Powoduje problemy ze spójnością danych w pamięci cache i RAM

Algorytmy zapisu http://kik.pcz.pl/soold/mainpage/subject22_2/chapt2.html

Big endian - od najbardziej znaczących bajtów, tak jest w starych prockach

Little endian -od najmniej znaczących, tak jest juz w nowszych intelach, amdkach, itp. Mimo ze kalkulator windowsowy pokaze wg big endian to procek juz woli sobie liczyc w little endian

O bajcie ~~ . Bajt jest najmniejszą jednostką obsługiwaną przez procesor. Coś jak z taksówką, ośmioosobowym mikrobusem. Nawet jeśli chcemy pojechać sami, to i tak siedem siedzeń pojedzie pustych. Spróbujmy poprosić taksówkarza żeby wymontował te zbędne i że zapłacimy tylko 1/8 ceny. Tak samo zareagował by procesor poproszony o przetworzenie jednego bitu.

<http://digitalforensics.pl/podstawy-analizy-danych/bajt/>