

Organizacja i architektura komputerów

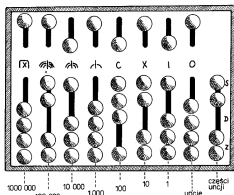
Wykład 2

Piotr Patronik

29 lutego 2016

Historia komputerów (1)

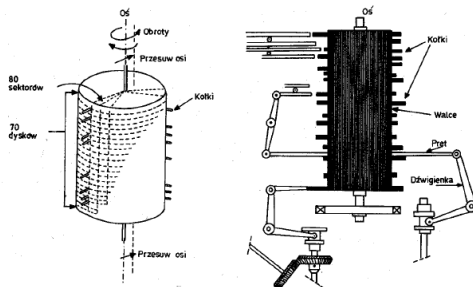
- ▶ system dwójkowy – Fou-Hi , Chiny 3000 lat p.n.e. (Yan – Yin)
- ▶ mozaika z Chow-Tse (Leibnitz, ok. 1702)
- ▶ mnożenie dwójkowe – Egipt 1650 p.n.e.(papiirus Ahmesa ok. 1560 p.n.e)
- ▶ abak (abacus) – czasy Cesarstwa Rzymskiego



- ▶ logarytm (1614) → pałeczki Nepera – 1617, mechanizacja mnożenia (system dziesiętny, pałeczka – wielokrotności liczb jednocyfrowych)

Historia komputerów (2)

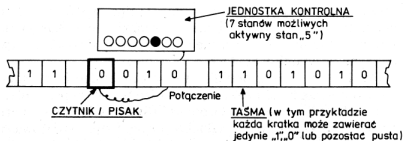
- ▶ maszyna różnicowa – Charles Babbage (1821-33), metoda różnic skończonych, realizacja – Edvard Scheutz (Szwecja, 1843)
- ▶ maszyna analityczna – Charles Babbage (1834) (możliwość programowania, Ada Lovelace) realizacja – Henry P. Babbage (1888), korekta wady – 1906



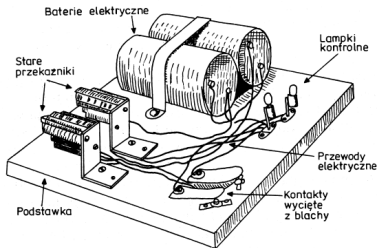
- ▶ reguły logiki dwuwartościowej – George Boole (1840)

Historia komputerów (3)

- ▶ maszyna algorytmiczna – Alan Turing, 1937

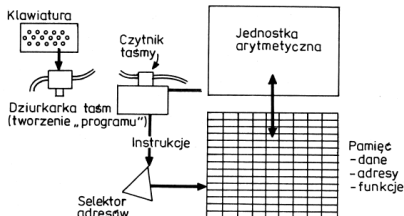


- ▶ przekątnikowe modelowanie struktur logicznych – Claude Shannon, 1937
- ▶ sumator przekątnikowy – George Stibitz (Bell Labs, 1937)



Historia komputerów (4)

- ▶ V-3 (Versuchmodel 3, Z3) – pierwszy komputer przekąźnikowy, Konrad Zuse (Niemcy, 1941) (V1–1936, V2–1938/1939), system dwójkowy, arytmetyka zmiennoprzecinkowa (bit ukryty)



- ▶ ABC – John Atanasoff, Clifford Berry – Iowa State University, 1941/42, pierwszy całkowicie elektroniczny kalkulator
- ▶ Colossus – zespół, Bletchley Park (W. Brytania), 1943/1945, pierwszy komputer elektroniczny, łamanie kodu Enigmy (przegląd zupełny)

Historia komputerów (4a)

- 1930 FET – Pierwsze patenty (J. E. Lilienfeld)
- 1951 Pierwszy (działający) tranzystor złączowy (W. Shockley)
- 1958 Pierwsze praktyczne układy scalone (J. Kilby, R. Noyce)
- 1966 Układ 7400 (TI)
- 1968 Int(egrated) El(ectronics) (G. E. Moore, R. Noyce)
- 1971 TMS-1000 (TI) Intel 4004 (Intel)

Historia komputerów (5)

- ▶ SCAS (Harvard MARK I) – Aiken 1944, elektromechaniczny kalkulator, minimalne możliwości realizacji decyzji, 1946 – warunkowe procedury, 1950 – wersja lampowa (MARK III),
- ▶ ENIAC (Electronic Numerator, Integrator And Calculator) – 1944/1946, J.P.Eckert, J.Mauchly (Moore School, Univ. Pennsylvania, pierwszy elektroniczny (18000 lamp), programowany przez tworzenie połączeń
- ▶ IAS – Princeton, pierwsza próba implementacji idei pamiętanego programu (J. von Neumann, 1945), realizacja 1951/52
- ▶ EDVAC (Electronic Discrete VArable Computer) – Cambridge (Anglia), M. Wilkes, 1949, pierwsza realizacja idei programu pamiętanego
- ▶ MU1 (Ferranti Mark 1) – Manchester (Anglia), T.Kilburn, 1951
- ▶ UNIVAC (UNIVersal Automatic Computer) – J.P.Eckert, przechowywany zakodowany program (1951), pierwszy komputer komercyjny
- ▶ PDP-1 – pierwszy mały komputer (DEC, 1960), PDP-8 – 1965
- ▶ Altair 8800 – komputer dla hobbystów (1975)
- ▶ Apple – komputer biurkowy, Steven Jobs, Stephen Wozniak (1976)

Etapy rozwoju (pokolenia) komputerów

- ▶ I generacja (1951-1960) – lampy elektronowe UNIVAC – Remington Rand (1951), IBM (1952) pamięć rdzeniowa (1951), tranzystor krzemowy (1954)
- ▶ II generacja (1960-1970) – tranzystory, magnetyczna pamięć rdzeniowa, PDP-1 (1960), minikomputer PDP-8 (1965) TTL (1964), pamięć SRAM (1968?)
- ▶ III generacja (1964-1975) – SSI, technologia TTL, IBM 360, systemy operacyjne tranzystor MOS (1970), pamięć DRAM (1974?)
- ▶ IV generacja (1974-1990) – VLSI, pamięci półprzewodnikowe mikroprocesory Intel 4004 (1971), Intel 8008 (1972) dyski magnetyczne, pamięć ferroelektryczna
- ▶ V generacja (1985-) – WSI, pamięci półprzewodnikowe procesory potokowe, wbudowana pamięć podręczna dyski optyczne, pamięć ferroelektryczna
- ▶ VI generacja (1995-) – komputery zintegrowane, komputery optyczne rozkazy typu SIMD (równoległość w zakresie instrukcji) dyski optyczne DVD, technologia blue-laser

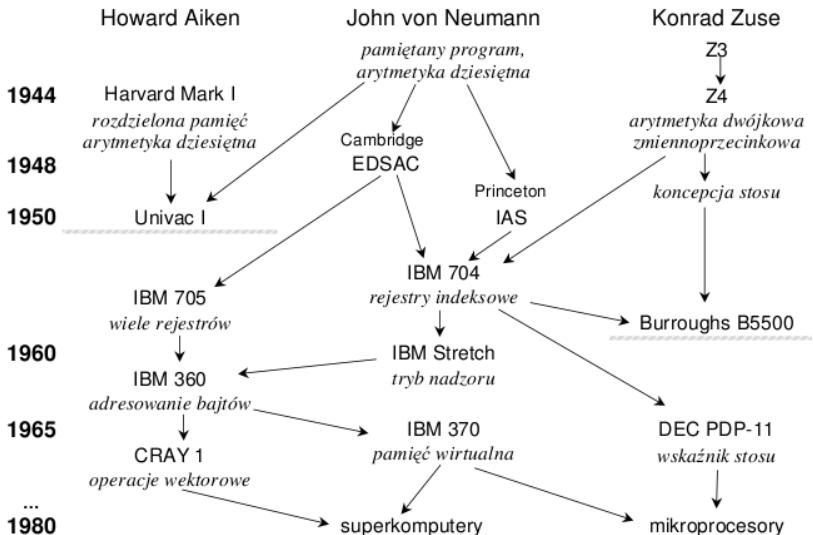
Komputery współczesne

► Evolucja architektury Pentium

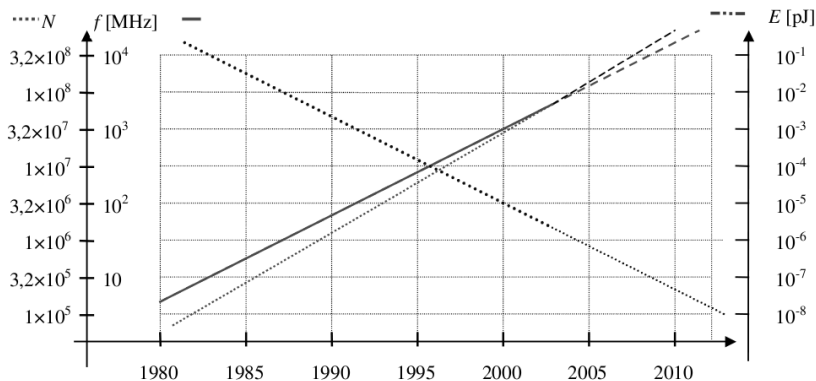
	Procesor	N_{TRANZ}	f_{max}	MIPS	adres	nowe cechy
1976	8085A	6,2 K	2MHz		64KB	
1978	8086 / 8088	29 K	10 MHz	0,8	1 MB	(koprocesor num. 8087)
1982	80286	134 K	12,5 MHz	2,7	16 MB	<i>tryb wirtualny 24b</i>
1985	386 DX / SX	375 K	33 MHz	6	4 GB	<i>tryb wirtualny 32b</i>
1989	486 DX / SX	1,2 M	66 MHz	20		cache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

- Architektura PowerPC/ARM – RISC
- Klastry komputerów –
przenośność oprogramowania – maszyna wirtualna Java
- Komputery wielordzeniowe/wielowątkowe
- Komputery przenośne, telefony, zegarki, ...
- Technologie łączności, sieci rozproszone, IoT, ...

Drzewo genealogiczne współczesnych komputerów



Tempo rozwoju technologii półprzewodnikowych



- ▶ N – liczba tranzystorów w układzie
- ▶ f – częstotliwość taktowania
- ▶ E – energia na działanie logiczne

Prawo Moore'a

- ▶ Tempo rozwoju technologii VLSI – obserwacja empiryczna
Co 18 miesięcy podwaja się gęstość upakowania układów VLSI (pojemność pamięci oraz osiągalna szybkość przełączania)
- ▶ Zależność wykładnicza: $V(t) = V_0 \cdot 2^{(t-t_0)/a}$

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

- ▶ II prawo Moore'a
Wzrost kosztu linii produkcyjnej dla nowych technologii powoduje utratę opłacalności wytwarzania

Rozwój technologii VLSI i prognoza

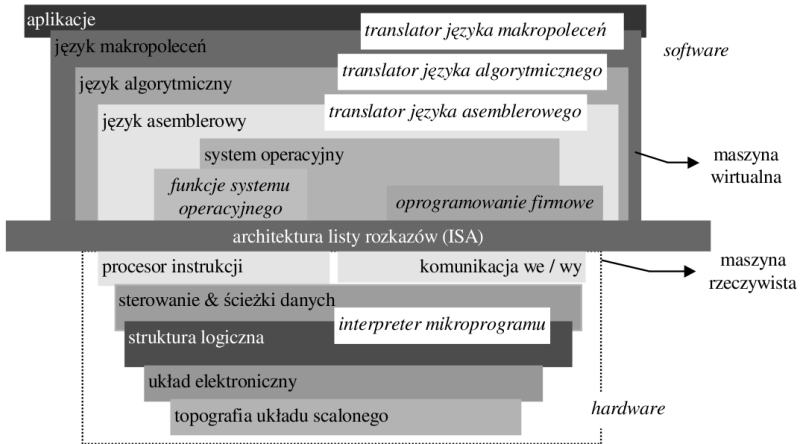
Rok	Grubość warstwy	Liczba warstw metalizacji	Napięcie zasilania	Częstotliwość
1985	1,6 μm	1	$\pm 5,0 \text{ V}$, 12,0 V	12 MHz
1990	1,0 μm	2	5,0 V	50 MHz
1995	0,5 μm	4	3,3 V	150 MHz
1998	0,25 μm	6	2,5 V	700 MHz
2000	0,15 μm	6	1,8 V	1 GHz
2005	0,08 μm	8	1,3 V	4 GHz
2010	0,06 μm	8	1,1 V	15 GHz

- Koszty wdrożenia rosną szybciej niż zyski ze sprzedaży

Hipoteza Landauera

- ▶ *stała Landauera* – minimalna energia przełączenia bitu
 $e = kT \ln 2 \approx 0,957 \cdot 10^{-11} T$ [pJ] $k \approx 1,38 \cdot 10^{-11} T$ [pJ/K]
– stała Boltzmannna
- ▶ W typowej temperaturze otoczenia $T \approx 315\text{K}$ (42°C) –
 $e \approx 3 \cdot 10^{-9}$ [pJ]
- ▶ Hipoteza Landauera – kres rozwoju technologii półprzewodnikowych
- ▶ Graniczny etap rozwoju: obniżenie energii przełączania do poziomu minimalnego

Poziomy maszynowe



- ▶ sprzężenia: powiązania nazw poziomu wyższego w identyfikatory na poziomie niższym
- ▶ identyfikatory: nazwy, adresy, lokacje

Poziomy maszynowe i zapis działań (1)

L4 – PASCAL

var A, B, C: integer;

....

A:=B+C;

L3 – Asembler (MC 680x0)

∞ A DS.W ; (adres A = 2002h)

∞ B DS.W ; (adres B = 2004h)

∞ C DS.W ; (adres C = 2006h)

∞ MAIN

∞ (*): MOVE.W B, D1

∞ (*+6): ADD.W C, D1

∞ (*+C): MOVE.W D1, A

L1 – kod maszynowy (MC 680x0)

0011	0010	0011	1000	(3238h)
0000	0000	0000	0000	(0000h)
0010	0000	0000	0100	(2004h)
1101	0010	0111	1000	(D278h)
0000	0000	0000	0000	(0000h)
0010	0000	0000	0110	(2006h)
0011	0001	1100	0001	(31C1h)
0000	0000	0000	0000	(0000h)
0010	0000	0000	0010	(2002h)

Poziomy maszynowe i zapis działań (2)

L4 – C++

integer A, B, C;

...

A=B+C;

L3 – Asembler (Intel x86)

∞ (.data)

∞ cc dw (?) ; (offset cc = 0103h)

∞ bb dw (?) ; (offset bb = 0105h)

∞ aa dw (?) ; (offset aa = 0107h)

∞ (.code)

∞ ...

∞ (*) mov ax, cc

∞ (*+3) add ax, bb

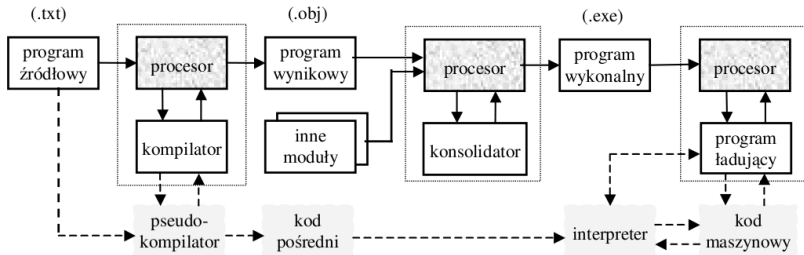
∞ (*+7) mov aa, ax

L1 – kod maszynowy (Intel x86)

1010	0001			(A1h)
0000	0011	0000	0001	(0103h)
0000	0011	0000	0110	(0306h)
0000	0101	0000	0001	(0105h)
1010	0011			(A3h)
0000	0111	0000	0001	(0107h)

► Składnia: Intel

Konwersja międzypoziomowa - kompilacja i interpretacja



Cechy języka ułatwiające kompilację

- ▶ *ortogonalność* zestawu działań i sposobów adresowania argumentów
 - ▶ *kompletność*: szeroki repertuar działań
 - ▶ *regularność*: jednorodność opisu działań
 - ▶ *oszczędność*: niewiele prostych rozkazów i sposobów adresowania
 - ▶ *dużo rejestrów* – łatwość przypisania rejestrów
- **cechy przejrzystej architektury**

Cechy przejrzystej architektury (1)

Spójność (ang. *consistency*)/regularność (ang. *regularity*)

► fragment \Rightarrow całość

► $\exists d \in \text{OP} \ \& \ \exists s \in \text{ARG}, (d, s) \in \text{ISA} \subset \text{OP} \times \text{ARG}$
 $\Rightarrow \forall f \in \text{OP}, \forall x \in \text{ARG} : (f, x) \in \text{ISA}$

add R1, R7, R15 \Rightarrow **add R1, R1, R10** \Rightarrow *add Rx, Ry, Rz*

\Downarrow

\Downarrow

\Downarrow

sub R1, R7, R15 \Rightarrow *sub R1, R1, R10* \Rightarrow *sub Rx, Ry, Rz*

\Downarrow

\Downarrow

\Downarrow

mul R1, R7, R15 \Rightarrow \Rightarrow *mul Rx, Ry, Rz*

\Downarrow

\Downarrow

\Downarrow

div R1, R7, R15 \Rightarrow \Rightarrow *div Rx, Ry, Rz*

spójność



ortogonalność

ogólność

trafność

Cechy przejrzystej architektury (1a)

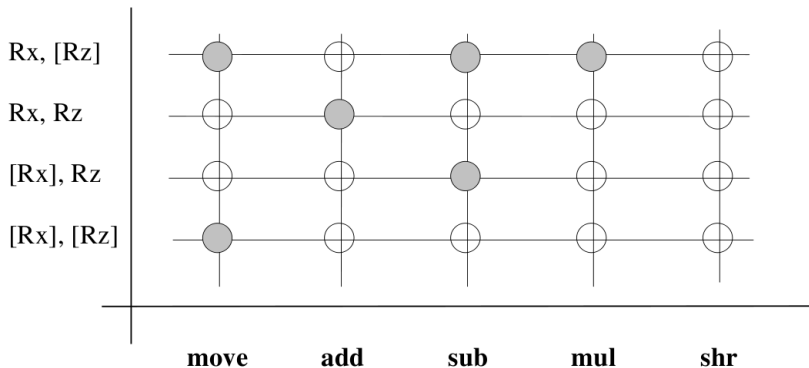
Regularność

- ▶ dostępność spodziewanych rozkazów i operandów
- ▶ dostępne dodawanie z przeniesieniem
 - ▶ dostępne odejmowanie z pożyczką
 - ▶ z takimi samymi zestawami operandów
 - z punktu widzenia regularności
- ▶ mnożenie dostępne jedynie z operandami rejestrowymi
 - ▶ dzielenie jedynie z operandami rejestrowymi
 - z punktu widzenia regularności

Cechy przejrzystej architektury (2)

Ortogonalność (ang. *orthogonality*)

- ▶ niezależność funkcji i operandów
- ▶ $\exists d \in \text{OP} : \forall x \in \text{ARG}, (d, x) \in \text{ISA}$
& $\exists s \in \text{ARG} : \forall f \in \text{OP}, (f, s) \in \text{ISA}$
 $\Rightarrow \forall f \in \text{OP}, \forall x \in \text{ARG} : (f, x) \in \text{ISA} \subset \text{OP} \times \text{ARG}$



Cechy przejrzystej architektury (2a)

Ortogonalność

- ▶ niektóre tryby adresowania pamięci dostępne z niektórymi rozkazami
- ▶ architektura ortogonalna
 - ▶ rozkaz odnoszący się do pamięci może używać dowolnego trybu adresowania
 - ▶ wybór tryb adresowania jest *ortogonalny* (= niezależny) od wyboru rozkazu
 - ▶ pary rozkaz/operand dostępne z takimi samymi trybami adresowania
- ▶ dodawanie z pewnymi trybami adresowania
 - ▶ odejmowanie (i inne operacje arytmetyczne i logiczne) z pewnymi trybami adresowania
 - z punktu widzenia ortogonalności
 - ▶ z tymi samymi trybami adresowania
 - z punktu widzenia regularności
- ▶ mnożenie z operandami rejestrowymi, a dzielenie nie
⇒ zbiór nie jest ortogonalny

Cechy przejrzystej architektury (3a)

Trafność (ang. *propriety*) (1)

- ▶ Przezroczystość (ang. *transparency*)
 - ▶ brak ograniczeń implementacyjnych
 - ▶ np. w rozmieszczeniu rozkazów w kodzie
 - ▶ np. w zakresie danych

Cechy przejrzystej architektury (3b)

Trafność (2)

► Oszczędność (ang. *parsimony*)

► $a, b, c, \dots p, q \in F \subset OP$

& $a = \Phi(p, q), b = \Gamma(p, q), c = \phi(p, q), \dots$

$\Rightarrow p, q \in OP$ & $a, b, c, \dots \notin OP$

<i>add Ra, Rb, Rc :</i>	sub Rx, Rx, Rx,	; Rx \leftarrow 0 (= Rx - Rx)
	sub Rb, Rx, Rb,	; Rb \leftarrow 0 - Rb
	sub Ra, Rb, Rc	; Rc \leftarrow Ra - (-Rb)

<i>and Ra, Rb, Rc :</i>	nor Ra, Ra, Ra,	; Ra \leftarrow ~ Ra
	nor Rb, Rb, Rb,	; Rb \leftarrow ~ Rb
	nor Ra, Rb, Rc	; Rc \leftarrow ~ (~ Ra \vee ~ Rb)

↓

\Rightarrow ISA:	sub Ra, Rb, Rc	; Rc \leftarrow Ra - Rb
\Rightarrow ISA:	nor Ra, Rb, Rc	; Rc \leftarrow ~(Ra \vee Rb)

Cechy przejrzystej architektury (4a)

Ogólność (ang. *generality*) (1)

► Kompletność (ang. *completeness*)

- sprzeczna z oszczędnością
- $d \in F, s \in A \text{ \& } (d, s) \in \text{ISA} \Rightarrow \forall f \in F, \forall x \in A : (f, x) \in \text{ISA}$
- spodziewane rozkazy i operandy
 - \Rightarrow dostępne
 - \Rightarrow z tym samym zbiorem trybów adresowania
- trudna do zdefiniowania
 - \rightarrow możliwość wykonania obliczeń w rozsądnym czasie
 - \rightarrow np. inkrementacja zamiast dodawania...
- niedostępność rozkazów z powodu oszczędności
 - \rightarrow rozkazy potrzebne muszą być dostępne, ale
 - \rightarrow np. mnożenie jest kosztowne sprzętowo... (AVR)
 - \rightarrow dzielenie tym bardziej

Cechy przejrzystej architektury (4b)

Ogólność (2)

- ▶ Otwartość (ang. *open-endedness*)
 - ▶ dostępna przestrzeń dla przyszłych rozszerzeń
 - ▶ regularna struktura kodu
 - wolne miejsca w przestrzeni kodowej
 - ▶ jednolite kodowanie argumentów
 - uniezależnienie działań od rozmiaru argumentów
 - ▶ konsekwentny dobór funkcji
 - unikanie ograniczeń implementacyjnych i technologicznych
 - ▶ implementacyjnych: nieograniczanie programisty
 - ▶ technologicznych: ...

Zagadnienia egzaminacyjne (1)

TEMATY EGZAMINACYJNE – OGÓLNE

1. Obsługa przerwań – priorytety, identyfikacja źródła
2. Ochrona procesu – zagrożenia, komunikacja
3. Obsługa i procesy wejścia i wyjścia
4. Obsługa wyjątków – klasyfikacja, wymagania
5. Zarządzanie pamięcią – cele, mechanizmy, wspomaganie na poziomie architektury
6. Wirtualna i rzeczywista przestrzeń adresowa; adres wirtualny, adres logiczny i fizyczny
7. Stronicowanie – adres wirtualny, tablice stron
8. Bufor antycypacji (TLB), odwrócone tablice stron
9. Segmentacja – segment wirtualny i rzeczywisty, translacja adresu, deskryptory (opisy segmentów)
10. Strategie przydziału stron i segmentów
11. Partycje i strategie wymiany stron – zbiór roboczy, zasada lokalności
12. Protokoły magistrali i arbitraż magistrali
13. Hierarchia pamięci – organizacja, budowa, obsługa
14. Zasada lokalności i jej zastosowania
15. Pamięć skojarzeniowa – definicja i organizacja
16. Pamięć podręczna – organizacja, charakterystyki
17. Bufory zapisu
18. Działania w pamięci podręcznej (wypełnianie, ...)
19. Strategie aktualizacji pamięci podręcznej
20. Antycypowane pobieranie linii, strategie wymiany
21. Spójność pamięci podręcznej – model MESI
22. Przetwarzanie potokowe – koncepcja, wymagania
23. Wydajność i przepustowość potoku
24. Organizacja potoku statycznego i dynamicznego
25. Konflikty przetwarzania w potoku i przestoje
26. Prognoza rozgałęzień – kolejka rozkazów, prognoza statyczna i dynamiczna
27. Konflikty strukturalne i ich usuwanie
28. Przemianowanie rejestrów
29. Stacje rezerwacyjne i buforowanie instrukcji
30. Komputer z programem zintegrowanym – definicja pamięci, rola licznika rozkazów
31. Architektura listy rozkazów – struktura kodu, koncepcja architektury RISC
32. Architektura procesora – R/M, R+M, L/S
33. Adresowanie danych – rozdzielczość adresowania
34. Separacja logiczna i fizyczna przestrzeni adresowych
35. Adresowanie bezpośrednie i pośrednie
36. Hierarchia i charakterystyka działań podstawowych
37. Struktury danych systemowych – stos i kolejka
38. Typy i formaty danych, charakterystyka
39. Arytmetyka stałoprzecinkowa – zasady, szybkość i poprawność działań, wykrywanie nadmiaru
40. Dodawanie i odejmowanie wielokrotnej precyzji – realizacja programowa
41. Arytmetyka zmiennoprzecinkowa – normalizacja, zaokrąglanie wyniku i cyfry chroniące, wyjątki
42. Przepływ sterowania – rozgałęzienia, warunki, instrukcje warunkowe
43. Przepływ sterowania – funkcje i procedury, blok aktywacji, okna rejestrowe
44. Model procesowy – stany istnienia procesu, konteksty, kontekst minimalny
45. Przełączanie i synchronizacja procesów – wzajemne wykluczanie, blokada

Zagadnienia egzaminacyjne (2)

WYMAGANE UMIEJĘTNOŚCI

1. Rozpoznawanie kodu stałoprzecinkowego
2. Konwersja kodów liczb całkowitych
3. Rozpoznawanie kodu zmiennoprzecinkowego
4. Analiza prostego programu asemblerowego
5. Odwzorowanie deklaracji danych w pamięci
6. Analiza działań arytmetycznych
7. Wykrywanie nadmiaru
8. Łamanie ograniczeń zakresu zmiennych stałoprzecinkowych
9. Symulacja działań zmiennoprzecinkowych
10. Obliczanie parametrów pamięci podręcznej – rozmiar linii, rozmiar bloku, pojemność
11. Ocena skuteczności pamięci podręcznej
12. Analiza stanu linii pamięci podręcznej (MESI)
13. Ocena trafności prognozy rozgałęzienia w danym algorytmie (programie)
14. Dekompozycja rozkazu na działania proste
15. Eliminacja rozgałęzień w programie
16. Obliczanie rozmiaru przestrzeni adresowych
17. Projektowanie deskryptora trybu wirtualnego
18. Ocena wydajności przetwarzania potokowego

PENTIUM / POWER PC

1. Adresowanie wirtualne w architekturze Pentium
2. Adresowanie wirtualne w architekturze PowerPC
3. Przerwania w architekturze Pentium
4. Przerwania w architekturze PowerPC
5. Tryb systemowy Pentium
6. Tryby adresowania Pentium i PowerPC
7. Tryby adresowania PowerPC
8. Jednostka zmiennoprzecinkowa Pentium i PowerPC
9. Kontekst zadania / procesu w Pentium
10. Kontekst zadania / procesu w PowerPC
11. Przełączanie zadań w Pentium, furtka
12. Przełączanie zadań w PowerPC, wywołanie systemowe
13. Sprzętowe wspomaganie ochrony danych w Pentium
14. Sprzętowe wspomaganie ochrony danych w PowerPC
15. Ewolucja architektury Pentium
16. Rozszerzenia MMX
17. Rozszerzenia SSE i SSE2
18. Identyfikacja procesora o architekturze Intel x86 – CPUID
19. Charakterystyka architektury Motorola 68K
20. Charakterystyka wybranej architektury RISC (SPARC, MIPS, ...)

Zadanie domowe

- ▶ Porównać modele programowe 8051 i AVR
 - ▶ Rejestry
 - ▶ Lista rozkazów
 - ▶ Tryby adresowania