Janusz Biernat, profesor Politechniki Wrocławskiej
Politechnika Wrocławska
Wydział Elektroniki
Katedra Informatyki Technicznej
(pracownia Architektury Komputerów)

ARCHITEKTURA KOMPUTERÓW 1

2018

☏ (+71) 320 3916 (… 2745)
janusz.biernat@pwr.edu.pl
⚒ p. 201 bud. C3

WYMAGANIA I PROGRAM

Krajowe ramy kwalifikacji (KRK)

Wydział Elektroniki PWr

KARTA PRZEDMIOTU

Nazwa w języku polskim: **Architektura komputerów 1/ Arytmetyka komputerów**

Nazwa w języku angielskim: **Computer Architecture 1 / Computer Arithmetic**

Kierunek studiów: Informatyka

Stopień studiów i forma: **I stopień, stacjonarna**

Rodzaj przedmiotu: **wybieralny/kierunkowy**

Kod przedmiotu: **INEK002 / INEK022**

Grupa kursów: **TAK**

	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium
Liczba godzin zajęć zorganizowanych w Uczelni (ZZU)	**15**	**30**			
Liczba godzin całkowitego nakładu pracy studenta (CNPS)	**60**	**75**			
Forma zaliczenia	ocena	ocena			
Kurs końcowy dla grupy kursów (X)	**X**				
Liczba punktów ECTS	**5**	**0**			
liczba punktów odpowiadająca zajęciom praktycznym (P)	**0**	**2,5**			
liczba punktów ECTS odpowiadająca zajęciom konsultacyjnym (BK)	**2,5**	**0**			

WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I INNYCH KOMPETENCJI

1. Zna podstawy arytmetyki pozycyjnej.
2. Rozumie pojęcie funkcji dyskretnej
3. Zna podstawowe funkcje logiczne i umie przekształcać wyrażenia logiczne.
4. Umie tworzyć proste algorytmy i projektować typowe struktury danych.
5. Potrafi przeprowadzić ciąg logicznego wnioskowania

CELE PRZEDMIOTU

C1. Nabycie wiedzy o arytmetyce uzupełnieniowej.
C2. Nabycie wiedzy o arytmetyce zmiennoprzecinkowej.
C3. Nabycie wiedzy o systemach arytmetyki resztowej i ich zastosowaniach
C4. Nabycie wiedzy z zakresu podstawowych algorytmów numerycznych.
C5. Nabycie umiejętności projektowania szybkich układów arytmetycznych.
C6. Nabycie umiejętności kontrolowania poprawności działań arytmetycznych.
C7. Nabycie umiejętności projektowania prostych algorytmów numerycznych.
C8. Nabycie umiejętności projektowania algorytmów arytmetyki rozszerzonego zakresu.

PRZEDMIOTOWE EFEKTY KSZTAŁCENIA

osoba, która zaliczyła kurs, ma następujące kompetencje:

z zakresu wiedzy:

PEK_W01 – zna zasady arytmetyki pozycyjnej i uzupełnieniowej

PEK_W02 – zna zasady arytmetyki zmiennoprzecinkowej

PEK_W03 – zna zasady arytmetyki resztowej

PEK_W04 – zna numeryczne algorytmy obliczania funkcji elementarnych.

PEK_W05 – zna podstawowe struktury układów arytmetycznych i rozumie ich działanie

z zakresu umiejętności:

PEK_U01 – umie wykonać działania arytmetyczne w arytmetyce uzupełnieniowej

PEK_U02 – umie wykonać działania arytmetyczne w arytmetyce zmiennoprzecinkowej

PEK_U03 – umie kontrolować poprawność działań arytmetycznych

PEK_U04 – potrafi zaprojektować układy arytmetyki uzupełnieniowej i zmiennoprzecinkowej

PEK_U05 – potrafi zaprojektować podstawowe układy arytmetyki resztowej.

PEK_U06 – potrafi zaprojektować struktury danych dla arytmetyki rozszerzonej precyzji i zakresu

z zakresu kompetencji społecznych:

PEK_K01 – potrafi poprawnie sformułować wypowiedź, tworzyć wnioski i opisywać obiekty

PEK_K02 – potrafi poprawnie wyciągać wnioski logiczne z faktów

TREŚCI PROGRAMOWE

	Forma zajęć - wykład	L. godzin
Wy1	Procesor i pamięć, dane i działania, adresowanie, warunki i rozgałęzienia. Reprezentacje liczb całkowitych: uzupełnieniowa, spolaryzowana oraz SD. Dodawanie i odejmowanie w systemach uzupełnieniowych, nadmiar.	2
Wy2	Konwersje podstawy systemu uzupełnieniowego. Wieloargumentowe dodawanie i algorytmy mnożenia w systemach uzupełnieniowych.	2
Wy3	Dzielenie odtwarzające i nieodtwarzające w systemach uzupełnieniowych. Obliczanie pierwiastka kwadratowego.	2
Wy4	Kongruencje, systemy resztowe, obliczanie reszt, algorytm Euklidesa. Chińskie twierdzenie o resztach (tw. Sun Tzu), twierdzenie Eulera.	2
Wy5	Standard IEEE754-2008. Algorytmy działań zmiennoprzecinkowych. Dokładność arytmetyki zmiennoprzecinkowej, metody zaokrąglania.	2
Wy6	Architektura układów arytmetycznych. Szybkie układy arytmetyczne	2
Wy7	Obliczenia przybliżone i obliczanie wartości funkcji elementarnych. Kontrola dokładności wyniku i arytmetyka wielokrotnej precyzji.	2
Wy8	Kolokwium zaliczeniowe	1
	Suma godzin	**15**

Forma zajęć – ćwiczenia		Liczba godzin
Cw1	Reprezentacje liczb całkowitych: uzupełnieniowa, spolaryzowana oraz SD.	2
Cw2	Dodawanie i odejmowanie w systemach uzupełnieniowych, nadmiar.	2
Cw3	Konwersje podstawy systemu naturalnego i uzupełnieniowego.	2
Cw4	Dodawanie wieloargumentowe i mnożenie w systemach uzupełnieniowych: algorytm Booth'a-McSorley'a, mnożenie bez rozszerzeń.	2
Cw5	Obliczanie pierwiastka kwadratowego.	2
Cw6	Dzielenie odtwarzające i nieodtwarzające w systemach uzupełnieniowych.	2
Cw7	Kongruencje, systemy resztowe, obliczanie reszt, algorytm Euklidesa.	2
Cw8	Chińskie twierdzenie o resztach (tw. Sun Tzu), twierdzenie Eulera.	2
Cw9	Architektura układów arytmetycznych.	2
Cw10	Szybkie układy arytmetyczne – sumatory PPA, układy i matryce mnożące	2
Cw11	Algorytmy działań zmiennoprzecinkowych i ich emulacja	2
Cw12	Dokładność arytmetyki zmiennoprzecinkowej, metody zaokrąglania.	2
Cw13	Obliczenia przybliżone i obliczanie wartości funkcji elementarnych.	2
Cw14	Kontrola dokładności wyniku i arytmetyka wielokrotnej precyzji.	2
Cw15	Testy zaliczeniowe	2
	Suma godzin	**30**

STOSOWANE NARZĘDZIA DYDAKTYCZNE

1. Wykład tradycyjny z wykorzystaniem wideoprojektora
2. Udostępnienie materiałów ilustracyjnych
3. Udostępnienie zbioru zadań i problemów wraz z sugestiami rozwiązania
4. Ćwiczenia rachunkowe
5. Konsultacje
6. Praca własna – samodzielne studia i przygotowanie do kolokwium

OCENA OSIĄGNIĘCIA PRZEDMIOTOWYCH EFEKTÓW KSZTAŁCENIA

Oceny (F – formująca, P – podsumowująca)	Numer efektu kształcenia	Sposób oceny osiągnięcia efektu kształcenia
F1	PEK_U01 ÷ PEK_U07	Odpowiedzi ustne, pisemne sprawozdania z ćwiczeń,
F2	PEK_W01 ÷ PEK_W05	Kolokwium pisemne
P = 0,5*F1 + 0,5*F2		

LITERATURA PODSTAWOWA I UZUPEŁNIAJĄCA

LITERATURA PODSTAWOWA

[1] BIERNAT J., Architektura układów arytmetyki resztowej, Warszawa, EXIT, 2007

[2] BIERNAT J., Architektura komputerów, Wrocław, Oficyna Wydawnicza PWr, 2005 (wyd. 4).

[3] KOREN I., Computer Arithmetic Algorithms, A.K.Peters, Natick, MA, 2002

LITERATURA UZUPEŁNIAJĄCA:

[1] BIERNAT J., Metody i układy arytmetyki komputerowej, Wrocław, Oficyna Wydawnicza PWr, 2001

[2] PARHAMI B., Computer Arithmetic. Algorithms and Hardware Designs, Oxford University Press, 2000

[3] WARREN H.S., Uczta programistów, Gliwice, Helion, 2003

[4] OMONDI A., PREMKUMAR B., Residue Number Systems, Imperial College Press, London, 2007

[5] http://www.zak.ict.pwr.wroc.pl/materials/architektura

[6] http://nandgame.com/

[7] J-M.MUELLER, *Elementary functions*, Boston, Birkhauser, 1997

OPIEKUN PRZEDMIOTU (IMIĘ, NAZWISKO, ADRES E-MAIL)

Janusz Biernat, 71 320 3916; janusz.biernat@pwr.edu.pl

„Dziś

wiemy więcej niż kiedykolwiek

ale

myślimy coraz mniej"

Neal Gabler, *New York Times*, 13.08.2011

ARCHITEKTURA KOMPUTERÓW 1

(ALGORYTMY ARYTMETYKI I UKŁADY CYFROWE)

Program wykładu (orientacyjny)

1. Systemy pozycyjne i uzupełnieniowe. Dodawanie, odejmowanie.
2. Mnożenie w systemie uzupełnieniowym i pozycyjnym.
3. Dzielenie w systemie uzupełnieniowym i pozycyjnym.
4. Obliczanie pierwiastka kwadratowego
5. Zapis liczby w żądanej podstawie. Schemat Hornera.
6. Szybkie mnożenie dwójkowe. Algorytm Bootha-McSorleya.
7. Dzielenie dwójkowe. Dzielenie nieodtwarzające.
8. Zmiennoprzecinkowa reprezentacja liczb.
9. Arytmetyka zmiennoprzecinkowa.
10. Sumatory warunkowe (COSA). Sumatory CLA i sumatory prefiksowe PPA.
11. Dodawanie wieloargumentowe. Reduktory CSA. Układy mnożące.
12. Chińskie twierdzenie o resztach i systemy resztowe.
    Algorytm Euklidesa, twierdzenie Carmichaela. Konwersja na i z RNS.
13. Elementarne metody numeryczne. [*)]
14. Arytmetyka wielkich liczb. [*)]

Program ćwiczeń 2018 (Z)

1. Systemy pozycyjne i uzupełnieniowe. Dodawanie, odejmowanie, liczby przeciwne.
2. Mnożenie i dzielenie. Obliczanie pierwiastka kwadratowego.
3. Reprezentacja liczby w systemie uzupełnieniowym. Schemat Hornera.
4. Algorytmy mnożenia dwójkowego. Algorytm Bootha-McSorleya.
5. Dzielenie dwójkowe odtwarzające i nieodtwarzające.
6. Reprezentacja zmiennoprzecinkowa i działania. Zaokrąglenia.
7. Logika układów cyfrowych. Sumatory dwuargumentowe i ich modyfikacje.
8. Propagacja przeniesień. Szybkie sumatory (CLA, PPA, COSA).
9. Dodawanie wieloargumentowe. Reduktory CSA. Układy mnożące.
10. Elementarne obliczenia numeryczne.
11. Algorytmy maszynowe podstawowych działań arytmetycznych. (C lub asembler gnu as)
12. Systemy resztowe. Algorytm Euklidesa, twierdzenie Carmichaela.
13. *Repetitio studiorum mater* .
14. Kolokwium.
15. Dogrywka.

WYMAGANE UMIEJĘTNOŚCI I WARUNKI ZALICZENIA

Wymagane umiejętności:

1. Konwersja kodów liczb całkowitych i ułamków
2. Wykonywanie działań na argumentach stało- i zmiennoprzecinkowych
3. Analiza i weryfikacja poprawności wykonania
4. Podstawy arytmetyki resztowej
5. Układowa implementacja działań
6. Analiza prostego programu asemblerowego

Warunki zaliczenia kursu:

– zaliczenie ćwiczeń:

– zaliczenie testu (algorytmy, implementacje)

Tematyka kolokwium

1. Arytmetyka stałoprzecinkowa – zasady, szybkość i poprawność działań, wykrywanie nadmiaru
2. Arytmetyka zmiennoprzecinkowa – normalizacja, zaokrąglanie wyniku i cyfry chroniące, wyjątki
3. Struktury układów cyfrowych – sumator dwu- i wieloargumentowy, układy mnożące
4. Asemblerowy zapis algorytmu i realizacja programowa działań wielokrotnej precyzji

ŹRÓDŁA WIEDZY

– **PODRĘCZNIKI**

– **http://www.zak.ict.pwr.wroc.pl/materials/architektura**

– **konsultacje**

ZASADY PRACY NA ĆWICZENIACH

– lista problemów/zadań: **http://www.zak.ict.pwr.wroc.pl/materials/architektura**
– kartkówki (zindywidualizowane: **PIN = PESEL/Nr_indeksu**)

0. Kapitał początkowy na każdych zajęciach: 2,5p → ocena za odpowiedź

1. Obowiązkowa znajomość rozwiązań bieżącej listy problemów/zadań

2. Nieprzygotowanie = –1p, ponowne = –2p

3. Nieobecność nieusprawiedliwiona = –2p

4. Kartkówka: – nieobecność lub brak rozwiązania (pustka kartka) = 0p

5. Obowiązkowy zeszyt zadań indywidualnych (32 k)

6. Premie:
– zgłoszenie do odpowiedzi (poprawnej): +1p
– wygrana w konkursie = ocena+2p

7. Kolokwium z umiejętności praktycznych w przedostatnim tygodniu zajęć
– warunek konieczny dopuszczenia do kolokwium: średnia osiągnięć ≥ 3.0p

Ocena końcowa – średnia oceny osiągnięć i wyniku kolokwium

ARYTMETYKA

Arytmetyka

Teoria liczb – właściwości liczb naturalnych

Arytmetyka – reprezentacja liczb (*struktura danych*)
– sposób wykonywania działań podstawowych (*algorytm*)

Arytmetyka klasyczna – dowolny rozmiar liczb (rozszerzenia nieskończone)

- ☝ *problem* – **przejrzysta reprezentacja liczb**
- ☝ *problem* – **wykonalność** obliczeń (wytworzenie poprawnego wyniku) **– algorytm**

Arytmetyka komputerowa – ograniczony zakres liczb

- ☝ *problem* – **reprezentacja liczb** dostosowana do ograniczeń
- ☝ *problem* – **algorytm** zapewniający
  - ☞ **wykonalność** obliczeń (wytworzenie poprawnego wyniku)
  - ☞ **dokładność** wyniku – utrzymanie dokładności
  - ☞ **poprawność** - kontrola zakresu
  - ☞ **szybkość** wykonania – zrównoleglenie działań

Arytmetyka – algorytmy

Elementarne działania arytmetyczne

- ***odejmowanie*** – działanie podstawowe
  - wytworzenie reprezentacji zera: 0=X–X
  - ***dodawanie*** wykonalne przez odejmowanie: X+Y=X–(0–Y)
  - wytworzenie reprezentacji liczby przeciwnej –X=0–X
- ***mnożenie*** – dodawanie skalowanych iloczynów częściowych
  - skalowanie – mnożenie przez całkowitą potęgę podstawy (bazy)
  - można wykonać sekwencyjne lub równoległe
- ***dzielenie*** – sekwencyjne odejmowanie (wielokrotności) dzielnika
- *obliczanie* ***pierwiastka kwadratowego*** – sekwencyjne poprawianie obliczonych przybliżeń pierwiastka

Obliczenia numeryczne

- *dzielenie przybliżone* – mnożenie przez odwrotność dzielnika
- *obliczanie odwrotności liczby i odwrotności pierwiastka kwadratowego*
- *obliczanie wartości funkcji elementarnych (przestępnych) – sin, tg, exp, ln,...*

REPREZENTACJE LICZB
I SYSTEMY LICZENIA

Optymalizacja: zapis pozycyjny

Jak za pomocą minimalnej liczby symboli zapisać dowolne liczby?

Rozwiązanie: **system pozycyjny** (podstawa, zbiór symboli (cyfr))
niezbędny symbol **zera**

Cyfry systemu dziesiętnego:

nazwa	zero	jeden	dwa	Trzy	cztery	pięć	sześć	siedem	osiem	dziewięć	używane
arabskie	0	1	2	3	4	5	6	7	8	9	świat
perskie	٠	١	٢	٣	٤	٥	٦	٧	٨	٩	Arabia
indyjskie	۰	۱	۲	۳	۴	۵	۶	۷	۸	۹	Indie

Zapis cyfr o wartościach większych od dziewięciu (powszechnie akceptowany)

wartość	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
symbol	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
binarnie	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
oktalnie	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17

Systemy pozycyjne (z ustaloną podstawą)

Systemy pozycyjne (ang. *fixed-radix, radix-based,* także *strictly positional*)

- *ustalona* ***podstawa*** (ang. *radix*) – zwykle liczba całkowita taka, że $|\beta|\geq 2$
- *waga pozycji* = całkowita potęga *podstawy*, *wykładnik*=numer pozycji, $w_i = \beta^i$
- reprezentacja liczby: wektor ***cyfr*** (ang. *digit*) o wartościach całkowitych, *wartość cyfry*=mnożnik wagi → niezbędny symbol „***zero***"
- *ustalony* zbiór wartości cyfr $\mathbf{D}=\{0,d_1,d_2,\ldots,d_{\beta-1},\ldots\}$, zawiera najmniej β wartości w tym 0, przy tym $d_i \bmod \beta \neq d_r \bmod \beta$ dla $i\neq r<\beta$, *standardowy* $\mathbf{D}=\{0,1,\ldots\beta-1\}$.
- *wartość liczby* = suma ważonych wartości cyfr:

Wartością X liczby o reprezentacji $\mathbf{X}=\{\ldots,x_{k-1},\ldots,x_1,x_0,\ldots,x_{-m},\ldots\}_\beta$, jest:

$$X=\ldots+x_{k-1}\beta^{k-1}+\ldots+x_1\beta+x_0+x_{-1}\beta^{-1}+\ldots+x_{-r}\beta^{-r}+\ldots=\sum_i x_i\beta^i$$

- *dokładność reprezentacji* = dolna granica sumowania → waga najniższej pozycji
- *zakres reprezentacji* = górna granica sumowania → *f*(waga najwyższej pozycji)

Skalowanie: $X=\sum x_i\beta^i=\beta^{-s}\sum x_{i-s}\beta^i$ (przesunięcie cyfr reprezentacji), liczba dana z dokładnością β^{-m} jest skalowaną liczbą całkowitą: $\sum_{i=-m}^{\infty} x_i\beta^i=\beta^{-m}\sum_{i=0}^{\infty} x_{i-m}\beta^i$

Jednorodna reprezentacja liczb dodatnich i ujemnych

W systemie naturalnym (ang. *natural*) – podstawa naturalna, cyfry dodatnie – można zapisać tylko liczby dodatnie.

Reprezentacje pozycyjne liczb dodatnich i ujemnych

- ***z cyframi znakowanymi*** (ang. *signed digit, SD*) – cyfry całkowite
    - dozwolone są *ujemne wartości cyfr*, np. $\underline{\mathbf{D}}=\{\ldots, \underline{2}, \underline{1}, 0, 1, \ldots\} \mid |\underline{\mathbf{D}}| \geq \beta\}$, *system nieredundantny*: $\underline{\mathbf{D}}_{\beta}=\{d_i\colon d_i=i \vee i-\beta,\ d_0=0\}$, np. $\underline{\mathbf{D}}_{10}=\{0,1,\underline{8},3,4,\underline{5},\underline{4},7,\underline{2},\underline{1}\}$
- ***uzupełnieniowa*** (ang. *radix-complement*) – rozszerzenie systemu naturalnego
    - liczbę $-X$ przeciwną do danej X reprezentuje wynik działania pozycyjnego $0-X$ jako nieskończony ciąg pozycyjny, możliwa reprezentacja skrócona,
- *z ujemną podstawą* (ang. *negative radix*) $\beta \leq -2$, *standardowy zbiór cyfr* $\mathbf{D}=\{0,1,\ldots,\beta-1\}$
    - *duża asymetria, skomplikowana arytmetyka – podwójne przeniesienia*
- *obciążona, +N* (ang. *biased N, excess N*) – naturalna reprezentacja liczby całkowitej zwiększonej o stałą naturalną N,
    - *spolaryzowana dodatnio*, $N=\frac{1}{2}\beta^k$ lub *spolaryzowana ujemnie*, $N=\frac{1}{2}\beta^k-1$
    - tylko liczby całkowite, ograniczony zakres

Łączność i przemienność dodawania w zapisie pozycyjnym

73860,057	$+0\cdot10^{5}$	$+7\cdot10^{4}$	$+3\cdot10^{3}$	$+8\cdot10^{2}$	$+6\cdot10^{1}$	$+0\cdot10^{0}$	$+0\cdot10^{-1}$	$+5\cdot10^{-2}$	$+7\cdot10^{-3}$	$+0\cdot10^{-4}$
+58172,035	$+0\cdot10^{5}$	$+5\cdot10^{4}$	$+8\cdot10^{3}$	$+1\cdot10^{2}$	$+7\cdot10^{1}$	$+2\cdot10^{0}$	$+0\cdot10^{-1}$	$+3\cdot10^{-2}$	$+5\cdot10^{-3}$	$+0\cdot10^{-4}$
=										
=53872,055	$+0\cdot10^{5}$	$+5\cdot10^{4}$	$+3\cdot10^{3}$	$+8\cdot10^{2}$	$+7\cdot10^{1}$	$+2\cdot10^{0}$	$+0\cdot10^{-1}$	$+5\cdot10^{-2}$	$+5\cdot10^{-3}$	$+0\cdot10^{-4}$
+78160,037	$+0\cdot10^{5}$	$+7\cdot10^{4}$	$+8\cdot10^{3}$	$+1\cdot10^{2}$	$+6\cdot10^{1}$	$+0\cdot10^{0}$	$+0\cdot10^{-1}$	$+3\cdot10^{-2}$	$+7\cdot10^{-3}$	$+0\cdot10^{-4}$
=	$+0\cdot10^{5}$	$+12\cdot10^{4}$	$+11\cdot10^{3}$	$+9\cdot10^{2}$	$+13\cdot10^{1}$	$+2\cdot10^{0}$	$+0\cdot10^{-1}$	$+8\cdot10^{-2}$	$+12\cdot10^{-3}$	$+0\cdot10^{-4}$
=	$+1\cdot10^{5}$	$+2\cdot10^{4}$	$+0\cdot10^{3}$	$+9\cdot10^{2}$	$+0\cdot10^{1}$	$+2\cdot10^{0}$	$+0\cdot10^{-1}$	$+8\cdot10^{-2}$	$+0\cdot10^{-3}$	$+0\cdot10^{-4}$
+		$+1\cdot10^{4}$	$+1\cdot10^{3}$	$+1\cdot10^{2}$	$+3\cdot10^{1}$	$+0\cdot10^{0}$	$+0\cdot10^{-1}$	$+1\cdot10^{-2}$	$+2\cdot10^{-3}$	
=132032,092	$+\mathbf{1}\cdot10^{5}$	$+\mathbf{3}\cdot10^{4}$	$+\mathbf{2}\cdot10^{3}$	$+\mathbf{0}\cdot10^{2}$	$+\mathbf{3}\cdot10^{1}$	$+\mathbf{2}\cdot10^{0}$	$+\mathbf{0}\cdot10^{-1}$	$+\mathbf{9}\cdot10^{-2}$	$+\mathbf{2}\cdot10^{-3}$	$+\mathbf{0}\cdot10^{-4}$

Łączność i przemienność dodawania - dodawanie wieloargumentowe*)

83870,957	$+0\cdot10^{5}$	$+8\cdot10^{4}$	$+3\cdot10^{3}$	$+8\cdot10^{2}$	$+7\cdot10^{1}$	$+0\cdot10^{0}$	$+9\cdot10^{-1}$	$+5\cdot10^{-2}$	$+7\cdot10^{-3}$
83860,957	$+0\cdot10^{5}$	$+8\cdot10^{4}$	$+3\cdot10^{3}$	$+8\cdot10^{2}$	$+6\cdot10^{1}$	$+0\cdot10^{0}$	$+9\cdot10^{-1}$	$+5\cdot10^{-2}$	$+7\cdot10^{-3}$
83870,957	$+0\cdot10^{5}$	$+8\cdot10^{4}$	$+3\cdot10^{3}$	$+8\cdot10^{2}$	$+7\cdot10^{1}$	$+0\cdot10^{0}$	$+9\cdot10^{-1}$	$+5\cdot10^{-2}$	$+7\cdot10^{-3}$
83860,957	$+0\cdot10^{5}$	$+8\cdot10^{4}$	$+3\cdot10^{3}$	$+8\cdot10^{2}$	$+6\cdot10^{1}$	$+0\cdot10^{0}$	$+9\cdot10^{-1}$	$+5\cdot10^{-2}$	$+7\cdot10^{-3}$
64860,951	$+0\cdot10^{5}$	$+6\cdot10^{4}$	$+4\cdot10^{3}$	$+8\cdot10^{2}$	$+6\cdot10^{1}$	$+0\cdot10^{0}$	$+9\cdot10^{-1}$	$+5\cdot10^{-2}$	$+1\cdot10^{-3}$
95860,957	$+0\cdot10^{5}$	$+9\cdot10^{4}$	$+5\cdot10^{3}$	$+8\cdot10^{2}$	$+6\cdot10^{1}$	$+0\cdot10^{0}$	$+9\cdot10^{-1}$	$+5\cdot10^{-2}$	$+7\cdot10^{-3}$
93860,958	$+0\cdot10^{5}$	$+9\cdot10^{4}$	$+3\cdot10^{3}$	$+8\cdot10^{2}$	$+6\cdot10^{1}$	$+0\cdot10^{0}$	$+9\cdot10^{-1}$	$+5\cdot10^{-2}$	$+8\cdot10^{-3}$
16860,957	$+0\cdot10^{5}$	$+1\cdot10^{4}$	$+6\cdot10^{3}$	$+8\cdot10^{2}$	$+6\cdot10^{1}$	$+0\cdot10^{0}$	$+9\cdot10^{-1}$	$+5\cdot10^{-2}$	$+7\cdot10^{-3}$
+58172,935	$+0\cdot10^{5}$	$+5\cdot10^{4}$	$+8\cdot10^{3}$	$+1\cdot10^{2}$	$+7\cdot10^{1}$	$+2\cdot10^{0}$	$+9\cdot10^{-1}$	$+3\cdot10^{-2}$	$+5\cdot10^{-3}$
59872,955	$+0\cdot10^{5}$	$+5\cdot10^{4}$	$+9\cdot10^{3}$	$+8\cdot10^{2}$	$+7\cdot10^{1}$	$+2\cdot10^{0}$	$+9\cdot10^{-1}$	$+5\cdot10^{-2}$	$+5\cdot10^{-3}$
+78160,934	$+0\cdot10^{5}$	$+7\cdot10^{4}$	$+8\cdot10^{3}$	$+1\cdot10^{2}$	$+6\cdot10^{1}$	$+0\cdot10^{0}$	$+9\cdot10^{-1}$	$+3\cdot10^{-2}$	$+4\cdot10^{-3}$
=	$+0\cdot10^{5}$	$+76\cdot10^{4}$	$+55\cdot10^{3}$	$+74\cdot10^{2}$	$+70\cdot10^{1}$	$+4\cdot10^{0}$	$+99\cdot10^{-1}$	$+51\cdot10^{-2}$	$+65\cdot10^{-3}$
=	$+7\cdot10^{5}$	$+6\cdot10^{4}$	$+7\cdot10^{3}$	$+4\cdot10^{2}$	$+0\cdot10^{1}$	$+4\cdot10^{0}$	$+5\cdot10^{-1}$	$+1\cdot10^{-2}$	$+0\cdot10^{-3}$
+		$+5\cdot10^{4}$	$+5\cdot10^{3}$	$+7\cdot10^{2}$	$+0\cdot10^{1}$	$+9\cdot10^{0}$	$+9\cdot10^{-1}$	$+6\cdot10^{-2}$	$+6\cdot10^{-3}$
=823144,476	$+\mathbf{8}\cdot10^{5}$	$+\mathbf{2}\cdot10^{4}$	$+\mathbf{3}\cdot10^{3}$	$+\mathbf{1}\cdot10^{2}$	$+\mathbf{4}\cdot10^{1}$	$+\mathbf{4}\cdot10^{0}$	$+\mathbf{4}\cdot10^{-1}$	$+\mathbf{7}\cdot10^{-2}$	$+\mathbf{6}\cdot10^{-3}$

Zapis pozycyjny – rozdzielność mnożenia względem dodawania

$1947 \times 2674 = (\mathbf{1}\cdot10^3+\mathbf{9}\cdot10^2+\mathbf{4}\cdot10^1+\mathbf{7}\cdot10^0) \times (2\cdot10^3+6\cdot10^2+7\cdot10^1+4\cdot10^0) =$

$= 2\cdot10^6+6\cdot10^5+7\cdot10^4+4\cdot10^3+18\cdot10^5+54\cdot10^4+63\cdot10^3+36\cdot10^2+8\cdot10^4+24\cdot10^3+28\cdot10^2+16\cdot10^1+$
$+14\cdot10^3+42\cdot10^2+49\cdot10^1+28\cdot10^0 =$

$= (2)\cdot10^6+(6+18)\cdot10^5+(7+54+8)\cdot10^4+(4+63+24+14)\cdot10^3+(36+28+42)\cdot10^2 +$
$+(16+49)\cdot10^1+(28)\cdot10^0= \mathbf{2}\cdot10^6+\mathbf{34}\cdot10^5+\mathbf{69}\cdot10^4+\mathbf{105}\cdot10^3+\mathbf{106}\cdot10^2+\mathbf{65}\cdot10^1+\mathbf{28}\cdot10^0=$

$= \mathbf{2}\cdot10^6+(\mathbf{2}\cdot10+\mathbf{4})\cdot10^5+(\mathbf{6}\cdot10+\mathbf{9})\cdot10^4+(\mathbf{1}\cdot10^2+\mathbf{0}\cdot10+\mathbf{5})\cdot10^3+(\mathbf{1}\cdot10^2+\mathbf{0}\cdot10+\mathbf{6})\cdot10^2+$
$+(\mathbf{6}\cdot10+\mathbf{5})\cdot10^1+(\mathbf{2}\cdot10+\mathbf{8})\cdot10^0=$

$= (\mathbf{2}+\mathbf{2})\cdot10^6+(\mathbf{4}+\mathbf{6}+\mathbf{1})\cdot10^5+(\mathbf{9}+\mathbf{0}+\mathbf{1})\cdot10^4+(\mathbf{5}+\mathbf{0}+\mathbf{5})\cdot10^3+(\mathbf{6}+\mathbf{6})\cdot10^2+(\mathbf{5}+\mathbf{2})\cdot10^1+\mathbf{8}\cdot10^0=$

$= \mathbf{4}\cdot10^6+\mathbf{11}\cdot10^5+\mathbf{10}\cdot10^4+\mathbf{10}\cdot10^3+\mathbf{12}\cdot10^2+\mathbf{7}\cdot10^1+\mathbf{8}\cdot10^0=$

$= \mathbf{5}\cdot10^6+\mathbf{2}\cdot10^5+\mathbf{1}\cdot10^4+\mathbf{1}\cdot10^3+\mathbf{2}\cdot10^2+\mathbf{7}\cdot10^1+\mathbf{8}\cdot10^0=$

$= \mathbf{5\ 2\ 1\ 1\ 2\ 7\ 8}$

Komputerowe reprezentacje liczb

Kody *stałoprzecinkowe* (ang. *fixed point, radix-based*) – zapis pozycyjny
– izomorficzne z liczbami całkowitymi o podstawie β

ustalone położenie przecinka pozycyjnego (współczynnika skali m):

- *liczba* = *liczba całkowita* × β^{-m}

np. (m=3, β=8). $7145{,}123_8 = 7145123 \times 8^{-3}$, $0031{,}456_8 = 31456 \times 8^{-3}$,

Kody *zmiennoprzecinkowe* (ang. *floating point*) – zapis wykładniczy: złożenie pól

- *znak liczby* (ang. *sign*),
- *znacznik* (ang. *significand*) (mantysa (ang. *mantissa*)), w tym ułamek (ang. *fraction*)
- *wykładnik* (ang. *exponent*) potęgi podstawy (ang. *radix*) – podstawa stała

+3,27145123 E5 ($=3{,}27145123_{10} \times 10^5$), $-31415{,}9_{10} \times 10^{-4}$, $1{,}01001_2 \times 2^{1011}$

Reprezentacje resztowe (ang. *residue number system*, RNS)

- reprezentacja liczby – *wektor reszt* względem stałych bazy RNS
- tylko liczby całkowite

$$56_{\{2,3,5,7\}} = \{56 \bmod 2, 56 \bmod 3, 56 \bmod 5, 56 \bmod 7\} = \{0, 2, 1, 0\}$$

Reprezentacje *logarytmiczne* – znak | logarytm wartości bezwzględnej

Reprezentacje liczb ujemnych i dodatnich (1)

- ***reprezentacja znak-moduł*** (ang. *sign-modulus*)
  kod/symbol znaku | wartość bezwzględna reprezentowana pozycyjnie
  - mnożenie i dzielenie – jak w systemie naturalnym
  - podwójne zero – skomplikowane dodawanie i odejmowanie

- ***reprezentacja uzupełnieniowa*** (ang. *radix-complement*)
  definicja: (–X=X=0–X)
  reprezentacja liczby przeciwnej = wynik jej pozycyjnego odejmowania od 0
  - zachowane reguły odejmowania/dodawania pozycyjnego
  - reprezentacja systematyczna, łatwo rozszerzalna

- ***reprezentacja z użyciem cyfr znakowanych*** (ang. *signed digit*)
  reprezentacja pozycyjna z dopuszczeniem cyfr o wartości ujemnej
  - łatwa transformacja na uzupełnieniowy
  - skomplikowane kodowanie
  - algorytmy działań zależne od zbioru dozwolonych cyfr

Reprezentacje liczb ujemnych i dodatnich (2)

- ***reprezentacja +N (obciążona)*** (ang. *biased N, excess N*))
  reprezentacja pozycyjna wartości $X+N \geq 0$
  - sztywny zakres – brak możliwości rozszerzenia
  - łatwe porównanie, dodawanie i odejmowanie,
  - trudne mnożenie, bardzo trudne dzielenie
- ***reprezentacja spolaryzowana*** – reprezentacja +*N*, gdzie *N*=*połowa zakresu*, powiązana z reprezentacją uzupełnieniową
  - binarna spolaryzowana dodatnio, $N= 2^{k-1}-1$
  - binarna spolaryzowana ujemnie, $N= 2^{k-1}$

- ***inne możliwości zapisu***
  - ujemna podstawa (ang. *negative base*)
    - duża asymetria, trudna arytmetyka
  - zapis dopełnieniowy (ang. *digit-complement*), *pseudopozycyjny*
    - podwójne zero, skomplikowana arytmetyka

Reprezentacja uzupełnieniowa

Idea: *pominięcie ograniczenia wykonalności* pozycyjnego odejmowania ($\underline{1}$ = –1):

2735	…000 2735	99…99	…000 99…99
+7329	+…000 7329	+1	+…000 00…01
= 1 0064	= …001 0064	= 1 00…00	= …001 00…00
– 7329	– …000 7329	– 1	– …000 00…01
= 0 2735	= …000 2735	= 0 99…99	= …000 99…99

0064	= …000 0064	… (0)0064	00..00	= (0)00…00
– 7329	– …000 7329	… – (0)7329	– 1	– (0)00…01
= $\underline{1}$ 2735	= …999 2735	… = (9)2735	=$\underline{1}$ 99..99	= (9)99…99
+7329	+…000 7329	… + (0)7329	+1	+(0)00…01
0 0064	…000 2735	… = (0)2735	0 00..00	(0)00…00

Wnioski:

- reprezentacja ujemnej różnicy jest *arytmetycznie poprawna*
- wynik *odjęcia liczby od zera* jest poprawną reprezentacją ***liczby przeciwnej***
- pozycyjne *dodawanie i odejmowanie* takich reprezentacji jest poprawne
- użycie *nieskończonych rozszerzeń lewostronnych* jest *arytmetycznie poprawne.*

Reprezentacja liczby wymiernej w ustalonej podstawie

Reprezentacja stałoprzecinkowa – pozycyjna rozszerzona na dowolne ułamki

- każdą liczbę o **skończonym rozwinięciu pozycyjnym** można przedstawić jako iloczyn liczby całkowitej lub ułamka właściwego oraz potęgi β^s skali s :

$$\sum_{i=r}^{i=n-1} x_i \beta^i = \beta^r \sum_{i=0}^{i=n+r-1} x_{i+r} \beta^i = \beta^n \sum_{i=-1}^{i=r-n} x_{i+n} \beta^i$$

- ***skala*** musi być ***odwzorowana w algorytmie*** obliczeniowym

Reprezentacja wykładnicza/inżynierska/naukowa (zmiennoprzecinkowa): mnożnik – skala

- zmienna wartość współczynnika skali – notacja inżynierska/naukowa, np $3{,}14159 \cdot 10^3$, $6{,}02214179 \cdot 10^{23}$, $1{,}3806505(24) \cdot 10^{-23}$, 1,3806505(24)E-23
- możliwych wiele reprezentacji tej samej liczby
  $3{,}14159 \cdot 100$, $0{,}314159 \cdot 10^3$, $31415{,}9 \cdot 10^{-2}$, 314,159E-2
- w reprezentacji maszynowej (komputerowej) pożądana normalizacja
  $F = \pm M\beta^E$, warunek normalizacji: $\beta^{p-1} \leq |M| < \beta^p$

Binarne formaty zmiennoprzecinkowe

liczba znormalizowana (p=1 ⇒ wiodącym bitem mnożnika M jest zawsze 1)

$$F = (-1)^s 2^E (1+f), \quad 0 \le f < 1$$

brak reprezentacji zera !!

naturalną reprezentacją zera jest kod postaci ***s* 00...00 00...00**

liczba zdenormalizowana (ukryty bit „0") – kod wykładnika E_0: 00…0

$$F = (-1)^s 2^{E_0} (0+f), \quad 0 \le f < 1$$

wskazana równomierność pokrycia przedziału w pobliżu 0

$$2^{E_{\min}}(1+0) - 2^{E_0}(0+0{,}111\ldots1) \le 2^{E_{\min}}(1+0.00\ldots001)$$

Kody specjalne – **kod wykładnika: 11…1**

- (f=0) *nieskończoności* ±∞,
- (f≠0) *nie-liczby*, *NaN* – wyniki, które nie są liczbami

Format zmiennoprzecinkowy IEEE 754-2008

bit ukryty

s	*e e e … e e e e*	*b b b b … b b b b b b b b b b b b b*
znak 1 bit	E – wykładnik (+N) k bitów	f – część ułamkowa modułu znacznika $\lvert M \rvert$ m bitów

bin-32	(single)	k=8, m=23	$[s_{31} \,\|\|\, E_{30:23} \,\|\|\, f_{22:0}]$	$-126 \le E \le 127$
bin-64	(double)	k=11, m=52	$[s_{63} \,\|\|\, E_{62:52} \,\|\|\, f_{51:0}]$	$-1022 \le E \le 1023$
bin-128	(quadruple)	k=15, m=112	$[s_{127} \,\|\|\, E_{126:112} \,\|\|\, f_{111:0}]$	$-16382 \le E \le 16383$

Wykładnik	kod	Ułamek	Kod binarny	Wielkość
$E=-2^{k-1}+2$	$e\ldots ee=0\ldots 00$	f	$s\ 0\ldots 00\ b\ldots bb$	$F=(-1)^s(0+f)2^{E_0}$
$E_{\min} \le E \le E_{\max}$	$0\ldots 01 \le e\ldots ee \le 1\ldots 10$	f	$s\ e\ldots ee\ b\ldots bb$	$F=(-1)^s(1+f)2^E$
	$e\ldots ee=1\ldots 11$	$f=0$	$s\ 1\ldots 11\ 0\ldots 00$	$\pm\infty$
	$e\ldots ee=1\ldots 11$	$f\neq 0$	$s\ 1\ldots 11\ b..bb$	NaN

$E_{\min}=-2^{k-1}+2,\ \ E_{\max}=2^{k-1}-1$

Arytmetyka stałoprzecinkowa

Reprezentacja liczb – pozycyjna lub uzupełnieniowa

- nieograniczona dokładność reprezentacji
- nieograniczony zakres operandów

Dokładność obliczeń – dodawanie, odejmowanie, mnożenie

- argumenty dokładne – wynik także dokładny
- argumenty przybliżone – łatwa kontrola dokładności wyniku,
  – ryzyko kumulacji błędów przybliżeń

Właściwości

- zachowane *prawa łączności i przemienności* dodawania
- zachowane *prawo rozdzielności* mnożenia względem dodawania

Arytmetyka nasyceniowa

- jednakowe działania na wszystkich polach (rekordach) danych
- dyskryminacja (nasycanie) wyniku w ustalonych granicach
- ustalony zakres i dokładność
- zbędna kontrola poprawności wyniku

Arytmetyka zmiennoprzecinkowa (FPU)

Reprezentacja liczb – standard IEEE754-2008

- **ograniczona dokładność** reprezentacji
- reprezentacja niemal każdej liczby jest przybliżona
- **ograniczony zakres** operandów (format!)

Dokładność obliczeń – dodawanie, odejmowanie, mnożenie

$x \cdot y \neq y \circ x$

- konieczna normalizacja i zaokrąglanie wyniku
- wytworzony wynik działań zwykle niedokładny – zaokrąglenia
- ryzyko kumulacji błędów przybliżeń
- **nie są zachowane** prawa łączności i przemienności dodawania
- nie jest **zachowane** prawo rozdzielności mnożenia względem dodawania

Implementacja działań

- porównanie, dodawanie, odejmowanie, mnożenie – bezpośrednio
- dzielenie, obliczanie odwrotności i odwrotności pierwiastka kwadratowego obliczanie funkcji elementarnych – wykonywane numerycznie
- konieczna sygnalizacja wyjątków (stanów wymagających interwencji)