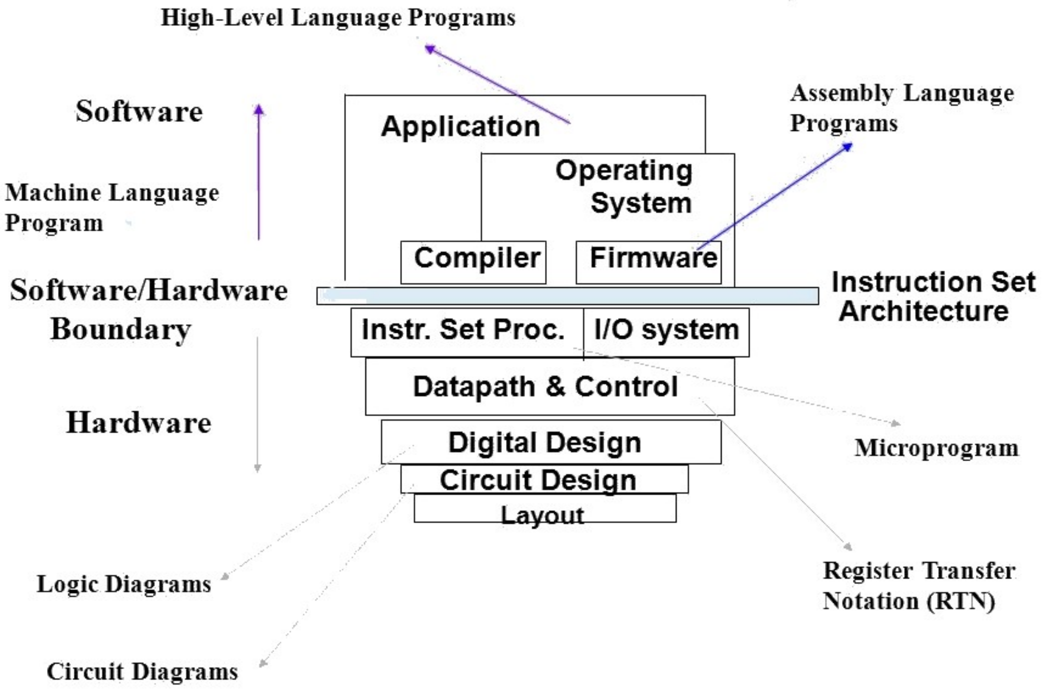
Architektury číslicových systémů

- zabývá se HW (s přesahem do SW) konstrukčními prvky číslicových systémů na všech úrovních (tranzistory, hradla, funkční bloky, aritmetické operace, sběrnice, řízení, procesory, instrukční sada, ..) jejich popisem (schema, stavový diagram, HDL (jazyky pro pro popis HW)) a algorimy pro návrh i realizaci.

- cíle návrhu číslicového systému – maximalizovat rychlost, výpočetní výkon, propustnost, … , minimalizovat složitost, plochu, spotřebu, ...

- naše oblast zájmu: Circuits design (návrh na úrovni hradel) – Instruction set Architecture (instr.sada)



Literatura

Tanenbaum, A. S.: Structured Computer Organisation, 4-6th Edition, Prentice Hall 1999-2013

Miloš D. Ercegovac, Tomás Lang, Digital Arithmetic, Morgan Kaufmann Publishers, 2004

Jean-Loup Baer: Microprocessor Architecture: From Simple Pipelines to Chip Multiprocessors, Cambridge University Press, 2009, ISBN 0521769922, 9780521769921

John L. Hennessy, David A. Patterson: Computer Architecture, Fifth Edition: A Quantitative Approach (The Morgan Kaufmann Series in Computer Architecture and Design), 2011, ISBN-13: 978-0123838728

Gook, M.: Hardwarová rozhraní, Průvodce programátora, Computer Press 2006

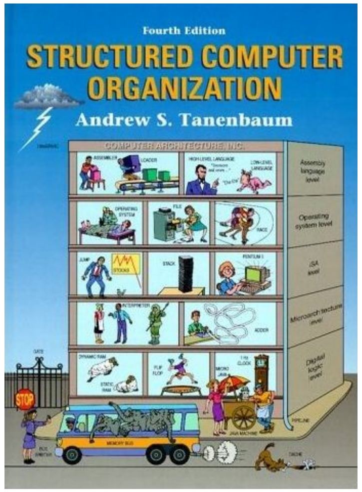
Pluháček A.: Computer Logic Design, skriptum ČVUT 2003

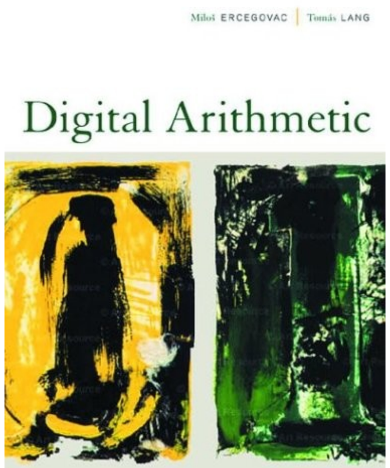
Douša J.: VHDL Language, skriptum ČVUT 2003

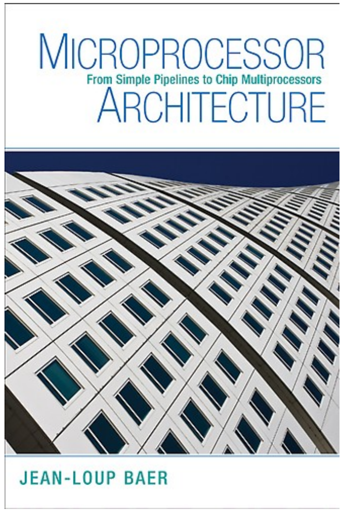
Pluháček, A.: Projektování logiky počítačů, ČVUT Praha, 2000

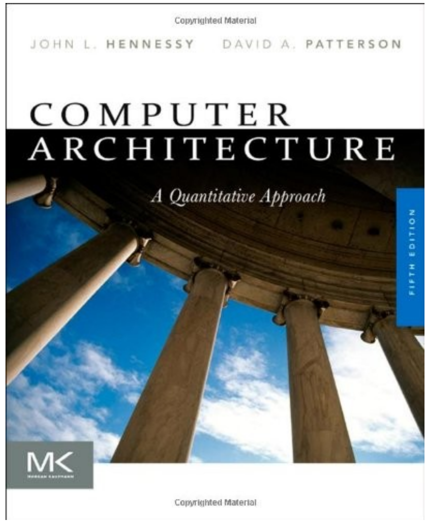
Jean-Pierre Deschamps, Gustavo D. Sutter, Enrique Cantó - Guide to FPGA Implementation of Arithmetic Functions

Jean-Michel Muller, Elementary Functions - Algorithms and Implementation









Číslicové systémy – Historie

Abakus (pomůcka) – starověké Řecko, Řím, Čína

– abakus – desticka vkládán kamínků (calculli)

Mechanické počítačí stroje

– Wilhelm Schickard, 1624: +, -, *, /

– Pascalina, 1642, vyr.50ks: +,-, pozdeji rozsirena o *,/

Děrné štítky - Joseph Marie, 1801 – tkalcovský stav kde vzorek látky bylo možné měnit změnou děrného šítku (ne přestavbou stroje). tj. první „program“

Programovatelné stroje

- Charles Babbage, 1833, „analytické stroj“ (mechanický) - první turingovsky úplný stroj

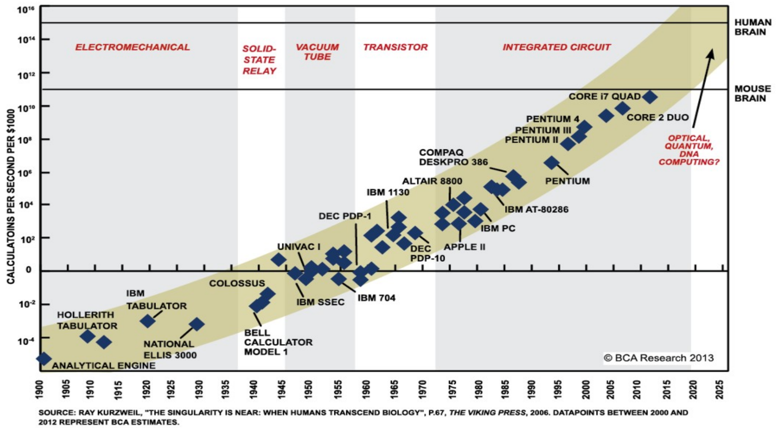
- Konrad Zuse (1934-1941) – reléový počítací stroj Z1,Z2,Z3 (200-2600relé)

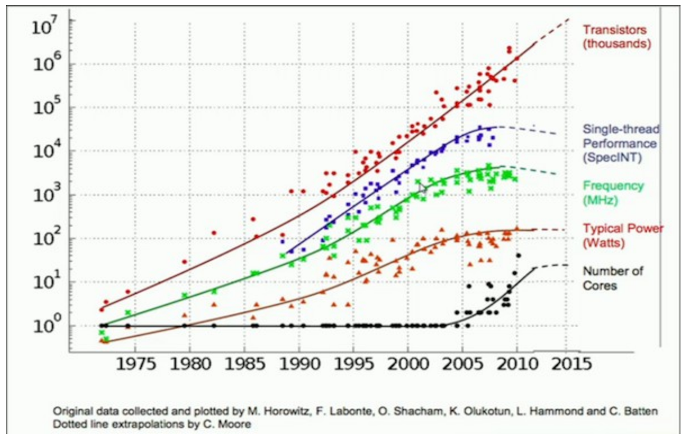
Elektronkové počítače (I.gen) - ENIAC, 1944 (cca 20000elektronek, 30tun, 63m2, 150kW)

Tranzistorové počítače (II.gen) – 1951-1965 – počátek oper.systémů, assembleru a prog.jazyků (cobol, fortran, algol)

Integrované obvody (III.generace) – 1965-1980 – počátek mutitaskingu, první minipočítače (předtím mainframy (sálové počítače))

Procesory (IV.generace) – od.r.1981 – rozvoj sítí (internet), mutiprocesorové systémy





Budoucnost (?)

Moorův zákon. Rychlost, spotřeba, cena.

Logické (digitální) obvody

- obvod pracuje s diskrétními stavy (vs.analogový), je tvořen log.členy(hradly). (rozhodovací úroveň, napěťové úrovně 0/1 In/Out)
- analogový - (+) rychlost, (-) přesnost, opakovatelnost, specializovanost) vs digitální - (+) opakovatelnost, univerzálnost
- kombinační (výstup fci vstupu) vs. Sekvenční (vnitřní stav) - Mealyho (Y=f(X,Z)), Moorův (Y=f(Z)
- asynchronní vs synchronní – úrovňové, hranové
- popis log.obvodu - výraz (logická funkce) , tabulka, Karnoughova mapa (K-mapa), log.obvod (schéma), diagram, program (HDL - Verilog, VHDL)
- Boolovská algebra – operátory a pravidla
- Základní log.fce (hradla) – NOT, AND/NAND, OR/NOR, XOR
- Základní Paměťové členy – Klopné obvody – RS, D, JK,
- Další stavební prvky – Buffery, drivery, registry, paměti, Dekodéry, Multiplexery, čítače
- Programovatelné log.obvody (PLD) – PAL/GAL, CPLD, FPGA
- procesory, mikrokontolery
- ASIC (zákaznické obvody – plně, polozákaznické)
- technologie – bipolární (e.g. TTL, IIL, ECL) , unipolární (PMOS, NMOS, CMOS)
- charakteristiky technologie – zpoždění(=rychlost), spotřeba, fanin (počet vstupů), fanout (počet připojitelných vstupů k výstupu), šumová odolnost, napájení, operační teplota, ...

Zobrazení čísel, aritmetika

- počítačová aritmetika – implementace aritmetických operací a funkcí – algoritmy pro HW i SW, kterou souvisí se zobr.čísel (přesnost, přetečení)

Zápis čísel

nejstarší 1:1 – záspis čísel počtem symbolů (6 mamutů = 6 kaménků), vytváření skupin po 5 nebo po 10

římský číselný systém – 1,5,10,50,100,500,1000 = I,V,X,C,L,D,M – nevhodný pro reprezentaci velkých a des.čísel, a pro aritm.operace

poziční číselný systém (poprvé v číně) – hodnota symbolu je dána polohou

smíšený poziční systém – základ není konstanta ale vektor, např. Čas H:M:S

poziční systém s pevným základem – základ R (radix) – typ.dekadická R=10 (123 = 1.10^2+2*10^1+3*10^0), **binární** R=2 (10=1*2^1+0*2^0)

optimální číselný základ (minimalizace stavů pro uchování informace) = e -> 3

vyvážená (symetrická) trojková číselná soustava – cifry {-1;0;1}, základ R=2, např. -2, -1, 0, 1,2,3,4, .. (dec) = $\bar{1}0$, $\bar{1}$, 0, 1, $1\bar{1}$, 10, 11, ..

redundatní systém se základem R=4 , cifry {-2, -1, 0, 1, 2}, napr. 6(dec) = 1 2 = 2 $\bar{2}$

systémy se záporným základem, systémy se zlomkovými základy, iracionální základy, základy s komplexním číslem,...

„*Ahhh, what an awful dream. Ones and zeroes everywhere…[shudder] and I thought I saw a two.*“ – Bender

„*It was just a dream, Bender. There's no such thing as two*“ – Fry

(Futurama)

Booleovská algebra

!x, -x, ~x, $\overline{x}$ = negace,inverze (~0=1, ~1=0)

x.y, x&y = log.soucin, and (0&0=0, 0&1=0, 1&0=0, 1&1=1)

x+y, x|y = log.součet, or (0+0=0, 0+1=1, 1+0=1, 1+1=1)

komutativita: x + y = y + x, x.y = y.x

distributivita: x + (y.z) = (x+y) . (x+z), x. (y+z) = (x.y) + (x.z)

neutralita 0/1: x+0 = x, x.1 = x

komplementarita: x + ~x = 1, x.~x = 0

(!pozor, + a . je jen symbol log.fcí OR a AND, není to arimetické + a *)

- asociativita: x + (y+z) = (x + y) + z, x.(y.z) = (x.y).z

- absorpce: x + (*x.y) = x*, *x.(x+y) = x*

- agresivita nuly,jedničky: x.0 = 0, x+1 = 1

- idempotence: x+x = x, x.x = x

- absorce negace: x+(~x.y) = x+y, x.(~x+y) = x.y

- dvojitá negace: ~(~x) = x

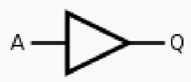
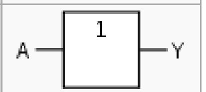
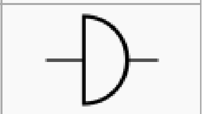
- komplementarita 0/1: ~0=1, ~1=0

- DeMorganovy zákony: ~x.~y = ~(x+y), ~x+~y = ~(x.y)

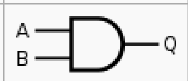
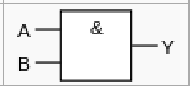
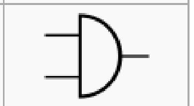
- DeMorganovy zákony: x.y = ~(~x+~y), x+y = ~(~x.~y)

Základní logické funkce (hradla)

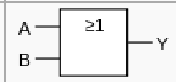
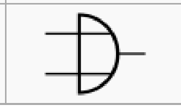
Opakovač, Repeater, Driver, Budič	AND, log.soucin, konjunkce	OR, log.součet, disjukce
NOT, Invertor, Negace	NAND, negovaný log.součin	XOR, exkluzivní log.součet

Funkce	$\mathbf{Y} = \mathbf{A}$
Značení	Pravdivostní tabulka
norma	symbol
ANSI/MIL	
IEC	
DIN	

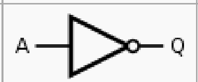
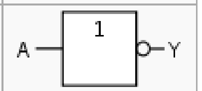
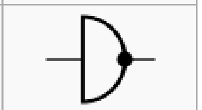
$X(A)$	Y
0	0
1	1

Funkce	$\mathbf{Y} = \mathbf{A} \cdot \mathbf{B}$
Značení	Pravdivostní tabulka
norma	symbol
ANSI/MIL	
IEC	
DIN	

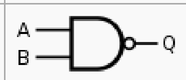
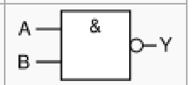
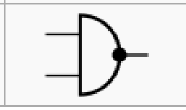
$X_1(A)$	$X_2(B)$	Y
0	0	0
0	1	0
1	0	0
1	1	1

Funkce	$\mathbf{Y} = \mathbf{A} + \mathbf{B}$
Značení	Pravdivostní tabulka
norma	symbol
ANSI/MIL	
IEC	
DIN	

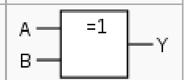
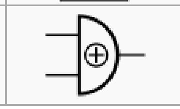
$X_1(A)$	$X_2(B)$	Y
0	0	0
0	1	1
1	0	1
1	1	1

Funkce	$\mathbf{Y} = \overline{\mathbf{A}}$
Značení	Pravdivostní tabulka
norma	symbol
ANSI/MIL	
IEC	
DIN	

$X(A)$	Y
0	1
1	0

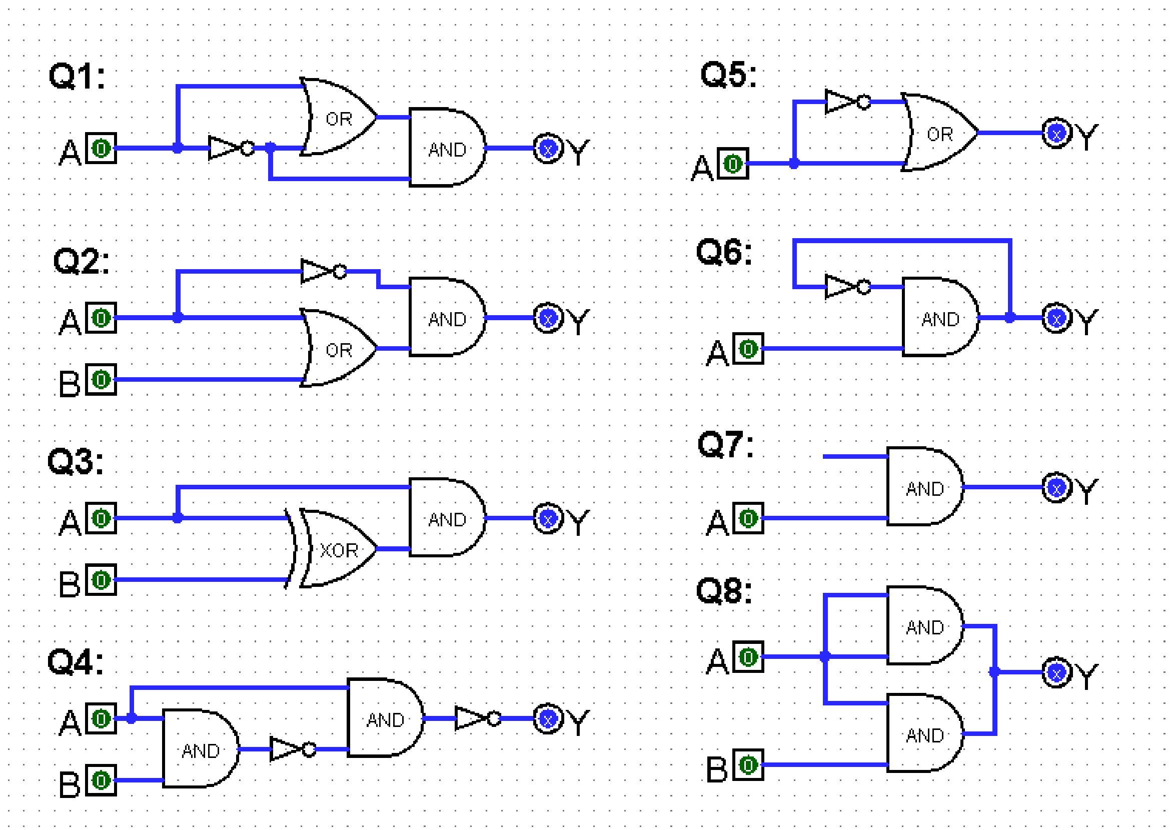
Funkce	$\mathbf{Y} = \overline{\mathbf{A} \cdot \mathbf{B}} = \overline{\mathbf{A}} + \overline{\mathbf{B}}$
Značení	Pravdivostní tabulka
norma	symbol
ANSI/MIL	
IEC	
DIN	

$X_1(A)$	$X_2(B)$	Y
0	0	1
0	1	1
1	0	1
1	1	0

Funkce	$\mathbf{Y} = \mathbf{A} \oplus \mathbf{B} = \overline{\mathbf{A}} \cdot \mathbf{B} + \mathbf{A} \cdot \overline{\mathbf{B}}$
Značení	Pravdivostní tabulka
norma	symbol
ANSI/MIL	
IEC	
DIN	

$X_1(A)$	$X_2(B)$	Y
0	0	0
0	1	1
1	0	1
1	1	0

Příklady (kresleno v programu logisim)



Q1,Q2,Q3,Q4: kdy je Y=1? (Q1: A=0; Q2: AB=01; Q3: AB=10; Q4: AB=00/01/11)

Q5: Y=?, platí stále? Pozor na přechodový jev → hazard (glitch)

Q6: Y=?. Pozor! kmitá pro A=1.

Q7: Y=?. Pozor! nedefinovaný vstup.

Q8: Y=?. Pozor! zkrat na výstupu

T.Mainzer - CPA

Zobrazení celých čísel (v binární poziční soustavě)

Kladná čísla – 1Byte – bit7..bit0, 0-255 (2^8), 2Byte 0-65535 (2^16), …

Znaménková čísla (1 Byte)

- Přímý kód – nejvyšší bit znaménko, 0=+, 1=-, tj. 0X84 = - 4.   je třeba testovat znaménko a podle toho sčítat odčítat, dvě nuly

- Inverzní kód – kladná čísla přímo, záporná inverzně, tj. -4 = -00000100 = 11111011.   stále dvě reprezentace nuly, jen myšlenkový přechod k:

- Doplňkový kód - kladná čísla přímo, záporná dvokový doplněk (negace a přičtení 1)   tj. -4 = - 00000100 = 11111011 +1 = 11111100

zobrazení -128 az 127 … 10000000 … 11111111 00000000 … 01111111

není dvojitá nula, vhodné při binární operace, přímá návaznost čísel -1 0 1   =snadná realizace aritmetiky (odčítání, sčítání)

- Kód s posunutou nulou

dohnutá „pozice“ 0, napr. 0=127. potom -4 = 123, 4=131. výhoda návaznosti, lepší porovnávání (vyšší číslo je vyšší), snadné sčítání

násobení je třeba ošetřit. Používá se pro reprezentaci exponentu pro reálná čísla

Příznakové bity, přetečení, přenos

Příznakové bity informují o výsledku operace ALU, např. když výsledek operace se nevejde do rozsahu. Příznakové bity:

Z = **Zero flag** - pokud je výsledek operace = 0, nastaví se na 1

C = **Carry flag** - Pokud došlo k **přenosu** do vyššího (MSB+1 = neexistujícího) bitu při operaci s unsigned číslem, nastaví se na 1 – tj. Hodnota MSB+1

V (O) = **Overflow** - Pokud došlo k **přetečení** u signed operace, nastaví se na 1. (tj. změna MSB)

N (S) = **Negative/Sign** - Pokud je výsledek záporný, nastaví se na 1 (kopie nejv.bitu ,MSB)

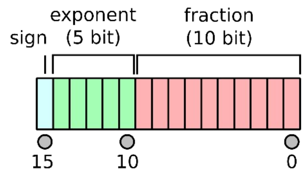
H = **Half Carry** - Pokud došlo k přenosu z nižších 4 bitů do vyšších, nastaví se na 1 (pro BCD)

Zobrazení reálných čísel

Fixní desetinná tečka

0001.1100 = 1.75 = 28/16

Exponenciální formát (IEEE 754)



FORMÁT: S Exp Mantisa   … S znaménko čísla (0=kladné, 1=záporné), Mantisa v Přímém kódu, Exponent v kód s posunutou nulou

Normalizovaný tvar Mantisy: 1.xxx .,První jedničku (která je tam vždy kromě výsledku 0) vynecháváme.

Exponent	**význam**
0 (00000)	0 or subnormal numbers
1-30 (00001-11110)	Normalised values
31 (11111)	Nan (infinity)

Příklad: float16 (IEEE 754, half-precision floating-point format)

Sign bit: 1 bit

Exponent: 5 bits (kod s pos.nulou. Offset = 15 = 01111), Emin=-14 = 00001, Emax = 15 = 11110, 00000/111111 reserved

Mantisa: 11 bits (10 is explicitly stored)

Binary	Hex	Value	Notes
0 00000 0000000000	0000	0	
0 00000 0000000001	0001	$2^{-14} \times (0 + \frac{1}{1024}) \approx 0.000000059604645$	smallest positive subnormal number
0 00000 1111111111	03ff	$2^{-14} \times (0 + \frac{1023}{1024}) \approx 0.000060975552$	largest subnormal number
0 00001 0000000000	0400	$2^{-14} \times (1 + \frac{0}{1024}) \approx 0.00006103515625$	smallest positive normal number
0 01101 0101010101	3555	$2^{-2} \times (1 + \frac{341}{1024}) \approx 0.33325195$	nearest value to 1/3
0 01110 1111111111	3bff	$2^{-1} \times (1 + \frac{1023}{1024}) \approx 0.99951172$	largest number < 1
0 01111 0000000000	3c00	$2^{0} \times (1 + \frac{0}{1024}) = 1$	one
0 01111 0000000001	3c01	$2^{0} \times (1 + \frac{1}{1024}) \approx 1.00097656$	smallest number >1
0 11110 1111111111	7bff	$2^{15} \times (1 + \frac{1023}{1024}) = 65504$	largest normal number
0 11111 0000000000	7c00	∞	infinity
1 00000 0000000000	8000	-0	
1 10000 0000000000	c000	-2	
1 11111 0000000000	fc00	$-\infty$	negative infinity

Přesnost, přetečení aritmetických operací

Celá čísla - číselný rozsah. Přenos (carry flag). Přetečení (overflow)

Pevná řádová čárka – číselný rozsah.

Pohyblivá řádová čárka – číselný rozsah/číselná osa. Přesnost (sčítání, násobení, zaokrouhlování)

a=1e10+1+1+1+1+1+1+...+1 //? int a, float a

a=10000*10000*10000*10000*10000; //? int a, float a

while(1) a+=1;   //? int a, float a

Důsledky chybné aritmetiky ...

Protiraketová střela Patriot v r.1991 – nepřesná kalkulace času – ochylka 0.34s/100hod → cca 500m odchylka (0.34s*1600m/s) → mimo dosah nepřátelské rakety -> nezničení, zabití 28lidí. Při testech krátký (několikahodinový) ‚up-time“ takže se neprojevilo.

Exploze rakety Ariane5 v r.1996 – 64bit float byl konvertován do 16 bit integer v modulu navigačnho systému (toto spolehlivě pracovalo v Ariane4,ale v Ariane5 byla tato hodnota 5* větší) → vyjímka mimo rozsah → 37s po vypuštění - exploze → přímá ztráta $500 000 000 (raketa a náklad), nepřímá $7 000 000 000 (vývoj)