Mikroprocesory ARM

–

zánik či naopak znovuzrození desktopu?

Pavel Tišnovský
<ptisnovs@redhat.cz>

Red Hat CZ

Obsah přednášky

- Základní informace o mikroprocesorech ARM
- Stručná historie architektury ARM
- Rodiny mikroprocesorů ARM
- Programátorský pohled na ARM
- Instrukční sada v režimu ARM
- Instrukční sada v režimu Thumb
- Využití mikroprocesorů ARM

Základní informace o mikroprocesorech ARM

ARM

- Acorn RISC Machine (1983)
- Advanced RISC Machines (1990)
- V současnosti název architektury procesorů RISC
- Jádro ARM součástí mnoha historických i současných čipů
  - „Od topinkovačů po servery“
  - Herní konzole
  - Tablety
  - Netbooky
  - Mobilní telefony
  - Postupná adaptace i na serverech

ARM + další rozšíření

- „Klasické“ procesory ARM
  - 32bitová instrukční sada RISC
- Další rozšíření
  - Instrukční sada Thumb (16 bitů)
  - Instrukční sada Thumb-2
  - Jazelle DBX (pro bajtkód JVM)
- MMU, GPU, ...

Tradiční obchodní model: Intel a AMD

- Tradiční model mainstreamových výrobců
- „Jedna velikost padne všem“
  - Relativně malé množství současně nabízených modelů CPU (10?)
- Jedna společnost (Intel či AMD)
  - Návrh architektury čipu
  - Výroba
  - Distribuce
- Poslední dobou změny ve firmě AMD

Obchodní model společnosti ARM

- Vlastní know-how a IP
- Nevyrábí vlastní procesory
- Namísto toho prodává IP dalším společnostem
  - Samsung
  - Qualcomm
  - NVidia
  - Nintendo
  - Texas Instruments
  - dalších cca 15 důležitých zákazníků

Přednosti architektury ARM

- Relativně malý počet tranzistorů
- Malá spotřeba (MIPS/Watt)
- Možnost získat licenci a zařadit jádro ARM do vlastního čipu
  - NVidia Tegra
  - OMAP
  - ...
- Dobrá podpora překladači
- Linux & ARM
  - Ekonomicky výhodné spojení technologií

Stručná historie architektury ARM

Projekt Acorn RISC Machine

- Acorn Computers Ltd.
  - Původní počítače založeny na MOS 6502

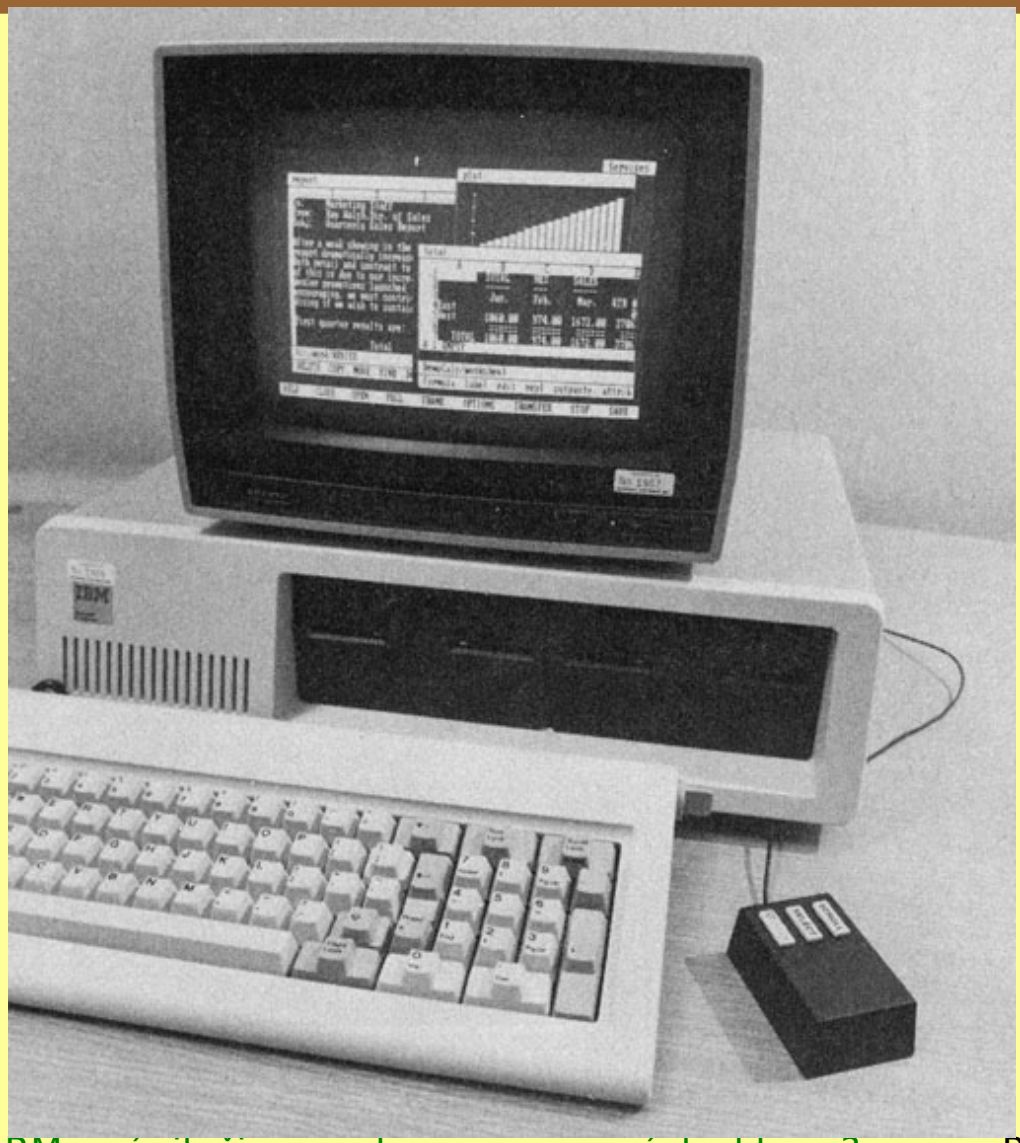
- Potřeba výkonnějšího a levnějšího CPU
  - Konkurence k PC (1981)
  - Později též konkurence k dalším počítačům
    - Atari ST
    - Amiga
    - Apple Macintosh

Amiga

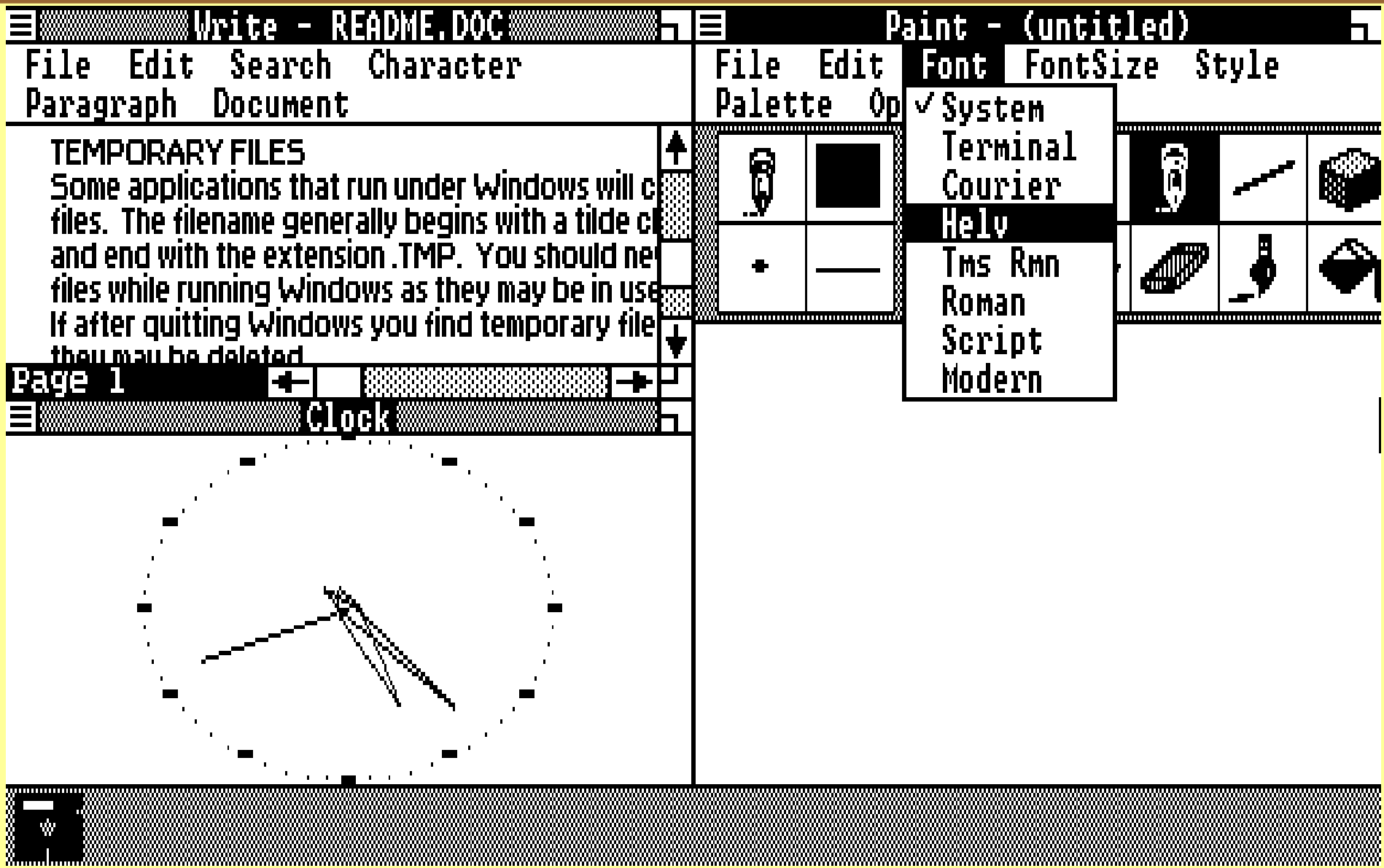
Atari ST

Macintosh

IBM PC + klony



??? (TM)


Write - README.DOC
File Edit Search Character
Paragraph Document
TEMPORARY FILES
Some applications that run under Windows will c
files. The filename generally begins with a tilde c
and end with the extension .TMP. You should ne
files while running Windows as they may be in use
If after quitting Windows you find temporary file
Page 1
Clock
Paint - (untitled)
File Edit Font FontSize Style
Palette Op
System
Terminal
Courier
Helv
Tms Rmn
Roman
Script
Modern

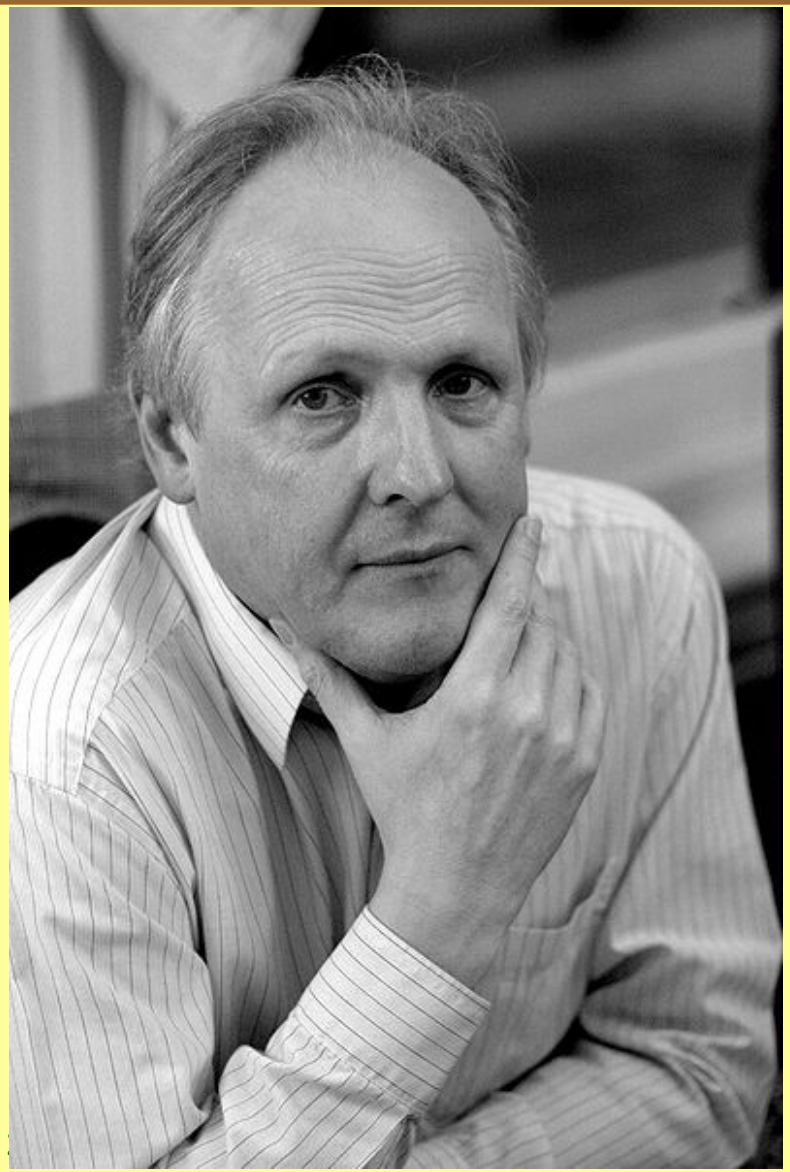
Výpočetní výkon čipů s architekturou CISC

- Intel 80286 @ 8 MHz
  - ~1,2 MIPS pro 16bitové operace
- Motorola 68000 @ 8MHz
  - ~1 MIPS pro 16bitové operace
  - ~1/2 MIPS pro 32bitové operace
  - ~2 MIPS maximum v syntetických testech

Projekt Acorn RISC Machine

- Steve Furber a Sophie Wilson
- Studie architektury Berkeley RISC
  - "Pokud skupinka studentů dokáže vytvořit plnohodnotný 32bitový procesor, nebude mít společnost Acorn vůbec žádný problém"
- RISC = nejenom "studentský CPU"

Steve Furber



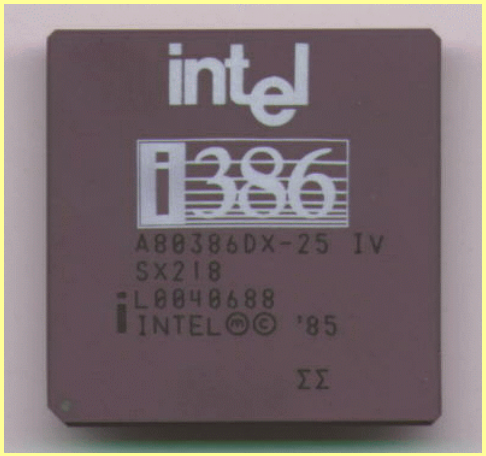
Sophie Wilson

Čipy ARM1/ARM2

- CPU bez mikrokódu a mikrořadiče
- Architektura Load/Store
- 32bit datová sběrnice
- 26bit adresová sběrnice
- PC jen 24 bitů, proto kód v 64MB prostoru (později 26b, nakonec 32b)
- 37 32bitových registrů
  - Některé mají speciální funkci
- Fixní délka operačních kódů 32bitů
- 1 instrukce/takt (kromě skoků a MUL)

Výsledek

- Intel 80286 @ 8 MHz
  - ~1,2 MIPS pro 16bitové operace
- Motorola 68000 @ 8MHz
  - ~1 MIPS pro 16bitové operace
  - ~1/2 MIPS pro 32bitové operace
  - ~2 MIPS maximum pro syntetické testy
- ARM2 @ 8MHz
  - ~4,5 MIPS pro 32bitové operace
- ARM3
  - výkonnostně překonal Intel 80286
  - I nejvyšší model @ 25 MHz



Výpočetní výkon však není vše

- Intel 80286
  - 134 000 tranzistorů
- Motorola 68000
  - 68000 tranzistorů
- ARM2
  - 30000 tranzistorů

Důsledky použití architektury RISC

- Počítače s ARM(2) se spoléhaly na hrubou výpočetní sílu CPU
- Orientace na 3D operace
  - x Amiga + Atari ST potřeba koprocesorů
  - BLITTER, Agnus, Denise...
  - Orientace na rastrové 2D operace

Výroba čipů ARM

- Společnost VLSI Technology, Inc.
- ARM1
  - ARM Evaluation System
  - 26.4.1985
  - první fungující čipy
  - říkalo se, že fungovala hned první „várka“, bez testování
  - bez HW násobičky
  - nebyly použity v komerčních počítačích
- ARM2
  - rok 1986

Acorn Archimedes

- Série domácích a osobních počítačů
- Acorn Computers Ltd
- CPU
  - ARM2
  - ARM3
  - ARM250
- RISC OS a RISC iX

BBC Archimedes 305

- 1987
- 512 kB RAM
- Mikroprocesor ARM2
- Rozhraní SCSI
- Cena od 899 liber

BBC Archimedes 305

Acorn Archimedes 410/1

- 1989
- 1 MB RAM
- Mikroprocesor ARM2
- Cena od 999 liber

Acorn Archimedes A3000

- 1990
- 1 - 2 MB RAM
- Mikroprocesor ARM2 či ARM250
- 4 - 8 MHz
- VIDC1a video čip
  - 160x256 pixelů, 4-256 barev
  - ...
  - 1056x250 pixelů, 16 barev
  - 1152x896 pixelů, 2 barvy
  - Barevná škála 4096 barev

Acorn Archimedes A3000

Acorn Archimedes A3000

Acorn A4

- Notebook
- jediný NB z celé řady počítačů Acorn
- 1992
- 2 - 4 MB RAM
- Mikroprocesor ARM3
- 24MHz
- 640x480 LCD (grayscale)
- Cena 1399/1699 liber

Acorn A4

Acorn A4

Rodiny mikroprocesorů ARM

Rodina/architektura/jádro

- Architektura
  - ARMv1 .. ARMv8
- Rodina
  - Svázána s architekturou
  - Odlišné číslování!
    - Arch:ARMv3: rodiny ARM6 a ARM7
- Jádro (Core)
  - Čipy se společnými moduly (cache, MMU, ...)
  - Příklad
    - ARMv3 => ARM 700, ARM 710 a ARM 710a

Rodina ARM1

- Architektura ARMv1
  - Jádro ARM1
  - První implementace jádra ARM
  - Bez HW násobičky
  - Bez MMU a bez cache
  - Proof of concept
    - Nebyl použit v žádném komerčním produktu

Rodina ARM2

- Architektura ARMv2
  - Jádro ARM2
  - První komerčně používané čipy ARM
- Novinky architektury ARMv2
  - Oproti ARM1 byla přidána HW násobička
- Základní parametry
  - Typická frekvence 8 MHz
  - Výkon přibližně 4 MIPS

Rodina ARM2 (pokr.)

- Architektura ARMv2a
  - Založeno na ARMv2 (původně jádro ARM2)
  - Nyní jádro A250
- Novinky architektury ARMv2a
  - Jádro A250
  - Integrovaná jednotka MMU
  - Nové instrukce
  - Grafický I/O procesor
- Základní parametry
  - Frekvence až 12 MHz
  - Výkon průměrně 7 MIPS

Rodina ARM3

- Stále architektura ARMv2a
- Jádro ARM3
- Postupně rostoucí hodinová frekvence
  - ARM250: 12 MHz
  - ARM3: 25 MHz
- Operační aměť se stává úzkým hrdlem architektury
  - Řešení typické pro RISC: použití cache
  - 4 kB cache v případě jádra ARM3 (D+I)
- Základní parametry
  - Frekvence až 25 MHz
  - Výkon průměrně 12 MIPS


C1
C4
C3
C2
C5
R1
U3
VLSI
ARM3PGA/ARM2 Adapter v2A
(C) Aleph One Ltd. 1991

Rodina ARM6

- Architektura ARMv3
- Novinky rodiny ARM6
  - Podpora pro plně 32bitové adresování (dříve 24, 26 bitů)
  - Volitelná cache podle jádra (0, 4 kB)
- Základní parametry
  - Frekvence až 33 MHz
  - Výkon průměrně 28 MIPS
- V některých zařízeních stále používán

Rodina ARM7

- Velmi úspěšná rodina čipů ARM
- Novinky rodiny ARM7
  - JTAG (ARM7DI)
    - Snazší ladění, hw breakpointy...
  - Thumb 16bit (u čipů s „T“ v názvu)
    - ARM7-TDMI
  - MMU (v závislosti na čipu)
  - Cache 8 kB (v závislosti na čipu)

Rodina StrongARM

- Architektura ARMv4
- Společnost DEC + Adv. ARM Machines
  - Později prodáno firmě Intel
    - Náhrada za i860 a i960
  - StrongARM → XScale
- Zaměření čipu
  - PDA
  - Set-top boxy
  - (Apple MessagePad 2000)
  - (Acorn Computers RISC PC)

SA-110

- První čip z rodiny StrongARM
- 1996
- Hodiny
  - 100, 160, 166, 200, 233 MHz
- Skalární architektura
- In-order
  - Méně tranzistorů
- RISCová pipeline s pěti řezy
- Bez branch prediction
  - Méně tranzistorů

SA-110

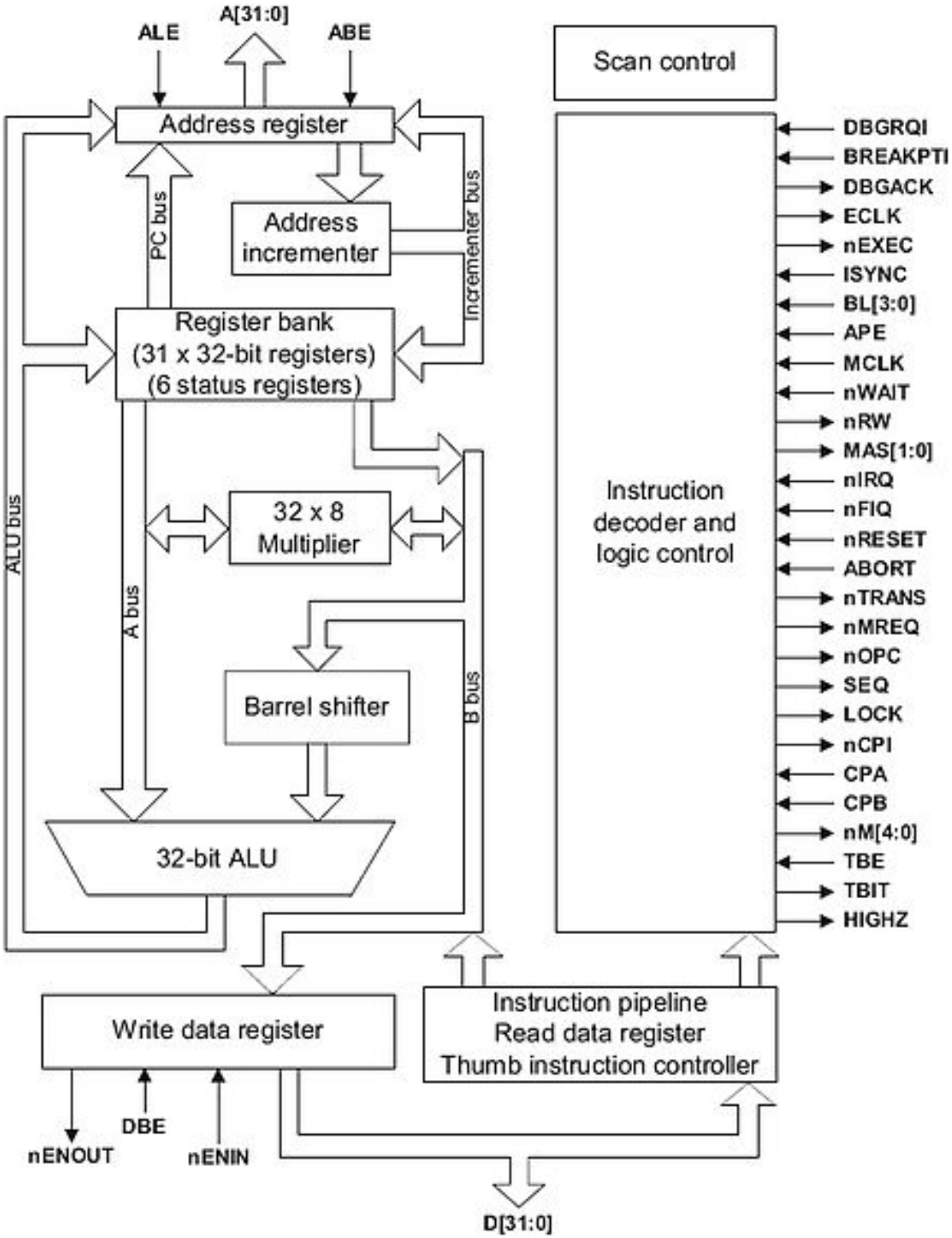
- Cache 16kB
- Technologie
  - CMOS
  - 0.35µm
- 2,5 milionu tranzistorů
- Napájení od 1,2 do 2,2V

Rodina StrongARM

SA-1100

- Založeno na SA-110
- Ještě užší zaměření na PDA
  - Integrace řadiče paměti
  - Řízení LCD a PCMCIA
  - Pět sériových I/O kanálů
    - USB
    - SDLC
    - UART (2x)
    - IrDA
- 2,5 milionu tranzistorů
  - Nutnost zmenšit cache na 8 kB ze 16 kB

Programátorský pohled na procesory ARM


ALE
A[31:0]
ABE
Address register
Address incrementer
PC bus
Incrementer bus
Register bank
(31 x 32-bit registers)
(6 status registers)
ALU bus
A bus
B bus
32 x 8
Multiplier
Barrel shifter
32-bit ALU
Write data register
Instruction pipeline
Read data register
Thumb instruction controller
nENOUT
DBE
nENIN
D[31:0]
Scan control
Instruction
decoder and
logic control
DBGRQI
BREAKPTI
DBGACK
ECLK
nEXEC
ISYNC
BL[3:0]
APE
MCLK
nWAIT
nRW
MAS[1:0]
nIRQ
nFIQ
nRESET
ABORT
nTRANS
nMREQ
nOPC
SEQ
LOCK
nCPI
CPA
CPB
nM[4:0]
TBE
TBIT
HIGHZ

Programátorský pohled na ARM

- Architektura typu Load-Store
  - Pro efektivní práci nutný velký počet registrů
  - Omezení přístupů do pomalé paměti
- Řešení
  - 37 registrů
  - Šířka registrů 32 bitů
  - Rozdělení do banků, které se překrývají
  - Pro každý stav procesoru zvláštní bank

Skupiny registrů

- 30 pracovních registrů (po 32 bitech)
- 1 program counter PC/r15
- 1 registr CPSR
  - Current Program Status Register
- 5 registrů SPSR
  - Saved Program Status Register

User32 / System	FIQ32	Supervisor32	Abort32	IRQ32	Undefined32
r0	r0	r0	r0	r0	r0
r1	r1	r1	r1	r1	r1
r2	r2	r2	r2	r2	r2
r3	r3	r3	r3	r3	r3
r4	r4	r4	r4	r4	r4
r5	r5	r5	r5	r5	r5
r6	r6	r6	r6	r6	r6
r7	r7	r7	r7	r7	r7
r8	r8_fiq	r8	r8	r8	r8
r9	r9_fiq	r9	r9	r9	r9
r10	r10_fiq	r10	r10	r10	r10
r11	r11_fiq	r11	r11	r11	r11
r12	r12_fiq	r12	r12	r12	r12
r13 (sp)	r13_fiq	r13_svc	r13_abt	r13_irq	r13_undef
r14 (lr)	r14_fiq	r14_svc	r14_abt	r14_irq	r14_undef
r15 (pc)	r15 (pc)	r15 (pc)	r15 (pc)	r15 (pc)	r15 (pc)

Program Status Registers

cpsr	cpsr	cpsr	cpsr	cpsr	cpsr
	spsr_fiq	spsr_svc	spsr_abt	spsr_irq	spsr_undef

Pracovní registry

- 30 pracovních registrů
- Z toho 15 viditelných v každém stavu
  - r0 .. r14
  - r13 se používá jako SP
  - r14 se používá jako LR (link register)

Program counter

- V assembleru PC či r15
- Zvyšován o hodnotu 4 v ARM režimu
- Zvyšován o hodnotu 2 v Thumb režimu
- Nastavován instrukcemi typu „branch“
- Nastavení registru r15 na hodnotu
  - Ve skutečnosti skok
- Návrat z podprogramu
  - MOV pc, lr

Current Program Status Register

- CPSR
  - Obsahuje kopie bitových příznaků získané z ALU
  - Režim práce procesoru
  - Příznak povolení/zákazu přerušení
  - Režim ARM/Thumb
    - Na některých jádrech
  - Režim ARM/Jazelle
    - Na některých jádrech
  - Příznak Q-saturace
    - Na některých jádrech

Saved Program Status Register

- SPSR
  - Kopie CPSR v rutině obsluhy výjimky
  - V každé rutině dostupný pouze jeden SPSR
  - V uživatelském režimu (user mode) nezajímavé

Stavy procesoru

- Neprivilegovaný
  - User mode
- Privilegované
  - FIQ - Fast Interrupt Request
  - IRQ - Interrupt Request
  - Supervisor
  - Abort
  - Undefined
  - Systém
    - ARM architecture v4 and above

Instrukční sada procesorů ARM

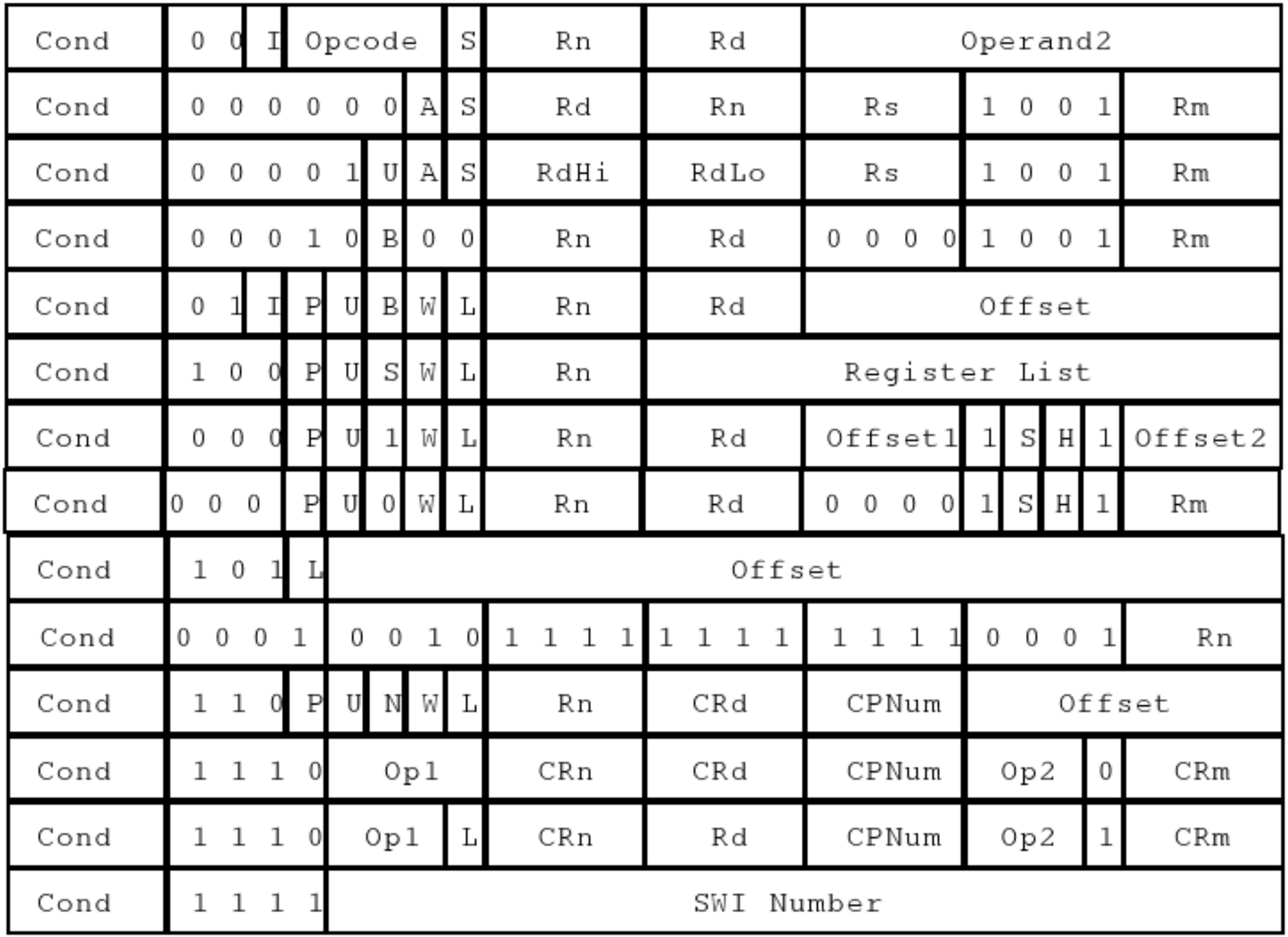
- Režim ARM
  - 32bitové instrukce
- Režim Thumb
  - 16bitové instrukce
- Režim Thumb2
  - Mix 16 a 32bitových instrukcí
- Režim Jazelle
  - Bajtkód JVM

Režim ARM

- Šířka všech instrukcí 32 bitů
- Šířka zpracovávaných dat taktéž 32 bitů
  - Problém s načítáním konstant
  - Musí se řešit na každé architektuře RISC

Typy instrukcí

- Load-store (jeden registr)
- Load-store (více registrů)
- Aritmetické operace
- Logické a bitové operace
- Skoky a změna režimu procesoru
- Práce se stavovými registry CPSR a SPSR
- Práce se semafory
- Instrukce koprocesoru(ů)


31 28 27 16 15 8 7 0
Cond 0 0 I Opcode S Rn Rd Operand2
Cond 0 0 0 0 0 0 A S Rd Rn Rs 1 0 0 1 Rm
Cond 0 0 0 0 1 U A S RdHi RdLo Rs 1 0 0 1 Rm
Cond 0 0 0 1 0 B 0 0 Rn Rd 0 0 0 0 1 0 0 1 Rm
Cond 0 1 I P U B W L Rn Rd Offset
Cond 1 0 0 P U S W L Rn Register List
Cond 0 0 0 P U 1 W L Rn Rd Offset1 1 S H 1 Offset2
Cond 0 0 0 P U 0 W L Rn Rd 0 0 0 0 1 S H 1 Rm
Cond 1 0 1 L Offset
Cond 0 0 0 1 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 Rn
Cond 1 1 0 P U N W L Rn CRd CPNum Offset
Cond 1 1 1 0 Op1 CRn CRd CPNum Op2 0 CRm
Cond 1 1 1 0 Op1 L CRn Rd CPNum Op2 1 CRm
Cond 1 1 1 1 SWI Number

Společné vlastnosti většiny instrukcí

- Spuštění na základě příznaků
- V mnoha případech není nutné používat skoky
  - Každý trvá 3 takty
- U mnoha instrukcí lze zvolit, které příznaky se mají nastavit
  - Výjimka CMP, CMN, TST a TEQ - nastaví příznaky vždy

Společné vlastnosti většiny instrukcí

- U mnoha instrukcí lze jeden operand posunout
  - Na vstupu ALU se nachází nezávislý barrel shifter
- Většina instrukcí přistupuje k r0..r14
- Některé instrukce taktéž k r15 (PC)

Příznaky nastavované ALU

- N - negative
- V - overflow
- Z - zero
- C - carry
- Q - sticky flag (v5e a výše)

Použití příznaků ALU v instrukcích

- MI N set < 0
- PL N clear >=0
- EQ Z set ==
- NE Z clear !=
- VS V set Overflow
- VC V clear No overflow
- AL Any Always
  - (většinou se nezapisuje, implicitní podmínka)

Použití příznaků ALU v instrukcích (pokr.)

- Porovnávání hodnot bez znaménka (unsigned)
- CS/HS C set >=
- CC/LO C clear <
- HI C set and Z clear >
- LS C clear or Z set <=

Použití příznaků ALU v instrukcích (pokr.)

- Porovnávání hodnot se znaménkem (signed)
- GE N and V the same >=
- LT N and V differ <
- GT Z clear, N == V >
- LE Z set, N != V <=

Příklad použití příznaků

```
int gcd(int a, int b)
{
    while (a != b) do
    {
        if (a > b) a = a - b;
        else       b = b - a;
    }
    return a;
}
```

Klasický přístup

```
gcd     CMP     r0, r1
        BEQ     end
        BLT     less
        SUB     r0, r0, r1
        B       gcd
less
        SUB     r1, r1, r0
        B       gcd
end
```

Práce kvalitního překladače/programátora

```
gcd
    CMP         r0, r1
    SUBGT       r0, r0, r1
    SUBLT       r1, r1, r0
    BNE         gcd
```

Výsledky

- Klasický přístup
  - 7 instrukcí
  - 28 bajtů
  - 13 taktů (branch=3 takty)
- Využití podmíněných instrukcí
  - 4 instrukce
  - 16 bajtů
  - 10 taktů (pouze jeden branch)
- Thumb
  - 7 instrukcí
  - 14 bajtů (16bit/instrukci)

Load-store (jeden registr)

- LDR
- STR
  - Podpora různých adresovacích režimů
    - relativní vůči PC
    - registr+offset
    - atd.
  - Různá šířka operandů
    - 32bitová slova (základ)
    - bajty (doplněny nulami či znaménkové rozšíření)
    - 16bitová slova (-//-)
    - 64bitová slova: dvojice registrů

Load-store (jeden registr)

- SWP
  - bajt nebo 32bitové slovo
  - implementace semaforů

Load-store (více registrů)

- LDM
- STM
  - Výběr kombinace registrů r0..r15

31	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Cond	0	0	I	Opcode				S	Rn				Rd				Operand2													
Cond	0	0	0	0	0	0	A	S	Rd				Rn				Rs				1	0	0	1	Rm					
Cond	0	0	0	0	1	U	A	S	RdHi				RdLo				Rs				1	0	0	1	Rm					
Cond	0	0	0	1	0	B	0	0	Rn				Rd				0	0	0	0	1	0	0	1	Rm					
Cond	0	1	I	P	U	B	W	L	Rn				Rd				Offset													
Cond	1	0	0	P	U	S	W	L	Rn				Register List																	
Cond	0	0	0	P	U	1	W	L	Rn				Rd				Offset1				1	S	H	1	Offset2					
Cond	0	0	0	P	U	0	W	L	Rn				Rd				0	0	0	0	1	S	H	1	Rm					
Cond	1	0	1	L	Offset																									
Cond	0	0	0	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	Rn					
Cond	1	1	0	P	U	N	W	L	Rn				CRd				CPNum				Offset									
Cond	1	1	1	0	Op1				CRn				CRd				CPNum				Op2			0	CRm					
Cond	1	1	1	0	Op1			L	CRn				Rd				CPNum				Op2			1	CRm					
Cond	1	1	1	1	SWI Number																									

Přesuny dat

- MOV
  - Přesun hodnoty registru/načtení konstanty
- MVN
  - Druhý operand je negován
- Lze použít i r15 jako cíl
  - Provede se skok
- Na druhý operand může být aplikována operace prováděná barrel shifterem:
  - ASR
  - LSL
  - LSR
  - ROR

Načítání konstant

- MOV
  - Osmibitová konstanta 0-255
  - Rotace bitů doleva
- MVN
  - Načte negovanou konstantu
- Řeší assembler automaticky
  - Pseudoinstrukce LDR
  - Dokáže použít buď MOV/MVN
  - Popř. konstantu z code segmentu

Aritmetické operace

- ADD
- ADC
- SUB
- SBC
- RSB (Reverse Subtract)
- RSC kombinace RSB a SBC

Aritmetické operace (pokr.)

- Na druhý operand může být aplikována operace prováděná barrel shifterem:
  - ASR
  - LSL
  - LSR
  - ROR

Aritmetika se saturací

- QADD
- QSUB
- QDADD
  - Druhý operand vynásoben 2
- QDSUB
  - Druhý operand vynásoben 2
- Pokud by mělo nastat přetečení, nastaví se bit Q

HW násobička

- Pracuje s trojicí registrů
- Instrukce typu „accumulate“ se čtveřicí registrů
  - Ještě se provádí součet

HW násobička (pokr.)

- MUL
- MLA
  - 32x32 => 32bitů (LSB)
- UMULL
- UMLAL
- SMULL
- SMLAL
  - 32x32 => 64 bitů

HW násobička (pokr.)

- SMULxy
- SMLAxy
  - 16x16 => 32 bitů
- SMULWy
- SMLAWy
  - 32x16 => 32bitů (MSB)
- SMLALxy
  - 16x16 => 64 bitů (accumulate)

Logické a bitové operace

- AND
- ORR
- EOR
- BIC
  - Bit Clear
  - AND s negovanými bity druhého operandu

Testovací instrukce (nastavují jen příznaky)

- CMP
- CMN
  - Compare negative
- TST
  - Provádí AND bez uložení výsledku
- TEQ
  - Provádí XOR bez uložení výsledku
- CLZ
  - Count Leading Zeroes
  - Vrací 32 pokud je zdroj=0
  - 0 pokud je nastaven nejvyšší bit na 1

Podmíněné skoky

- B label
  - Podmíněný skok (branch)
  - Rozsah skoku +-32MB
- BL label
  - Branch and link
  - Adresa další instrukce zkopírována do r14
  - Rozsah skoku +-32MB

Skoky a změna režimu procesoru

- BX Rm
  - Skok a změna instrukční sady!
  - Může provést přepnutí na Thumb
- BLX Rm
  - Kombinace BL a BX

Instrukce koprocesoru(ů)

- LDC
- STC

Přesuny dat mezi pamětí a koprocesorem

- MCR
- MCR2
- MCRR
- MRC
- MRC2

Přesuny dat mezi registry a koprocesorem

- CDP
- CDP2
- Specifické pro daný koprocesor

Thumb

- ARMv4T a další modely
- Šířka instrukcí 16bitů
- Podmnožina původní instrukční sady
- Optimalizováno s ohledem na překladače C a C++
- Přepínání mezi stavem ARM a stavem Thumb
  - Nutno přepínat, protože instrukce nejsou kompatibilní

Registry a Thumb

- Lo registers
  - RO-R7
- Hi registers
  - R8-R15
- Některé instrukce pracují pouze s jednou skupinou registrů

Větvení

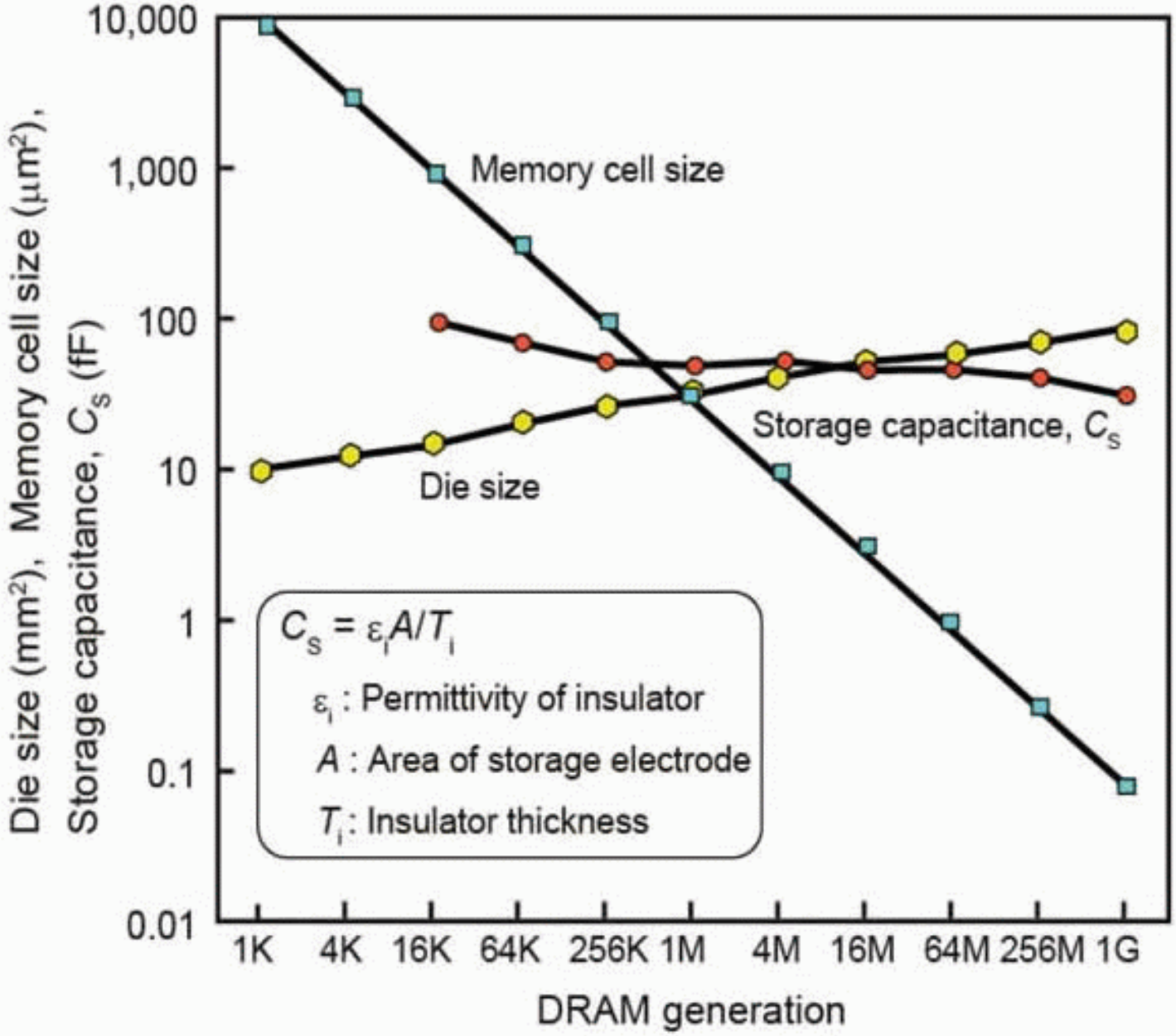
- Thumb neumožňuje podmíněné provádění instrukcí
- B[cond]
  - Branch
- CB[N]Z Rn
  - Compare and branch if (non) zero

Thumb 2

- Povoluje kombinace instrukcí o šířce 16 a 32 bitů
- Zpětná kompatibilita s Thumb
- Větší složitost čipů
- Větší „hustota“ strojového kódu
  - Srovnatelná s Thumb
- Má smysl ve světě rychlých CPU, pomalých DRAM a drahých cache
- Výkonnost srovnatelná s ARM režimem

Thumb 2

- IT
  - Instrukce odpovídající konstrukci if-then-else
- CBZ
  - Compare and Branch on Zero
- CBNZ
  - Compare and Branch on Non-Zero


10,000
1,000
100
10
1
0.1
0.01
Die size (mm²), Memory cell size (μm²),
Storage capacitance, C_s (fF)
Memory cell size
Storage capacitance, C_s
Die size
$C_s = \varepsilon_i A / T_i$
ε_i : Permittivity of insulator
A : Area of storage electrode
T_i : Insulator thickness
1K 4K 16K 64K 256K 1M 4M 16M 64M 256M 1G
DRAM generation

Jazelle

- Podpora pro spouštění většiny instrukcí JVM (bajtkód)
  - Proměnná délka instrukcí s osmibitovým opkódem a proměnným počtem operandů
- Musí existovat podpora v JVM
  - Jazelle Java Technology Enabling Kit (JTEK)

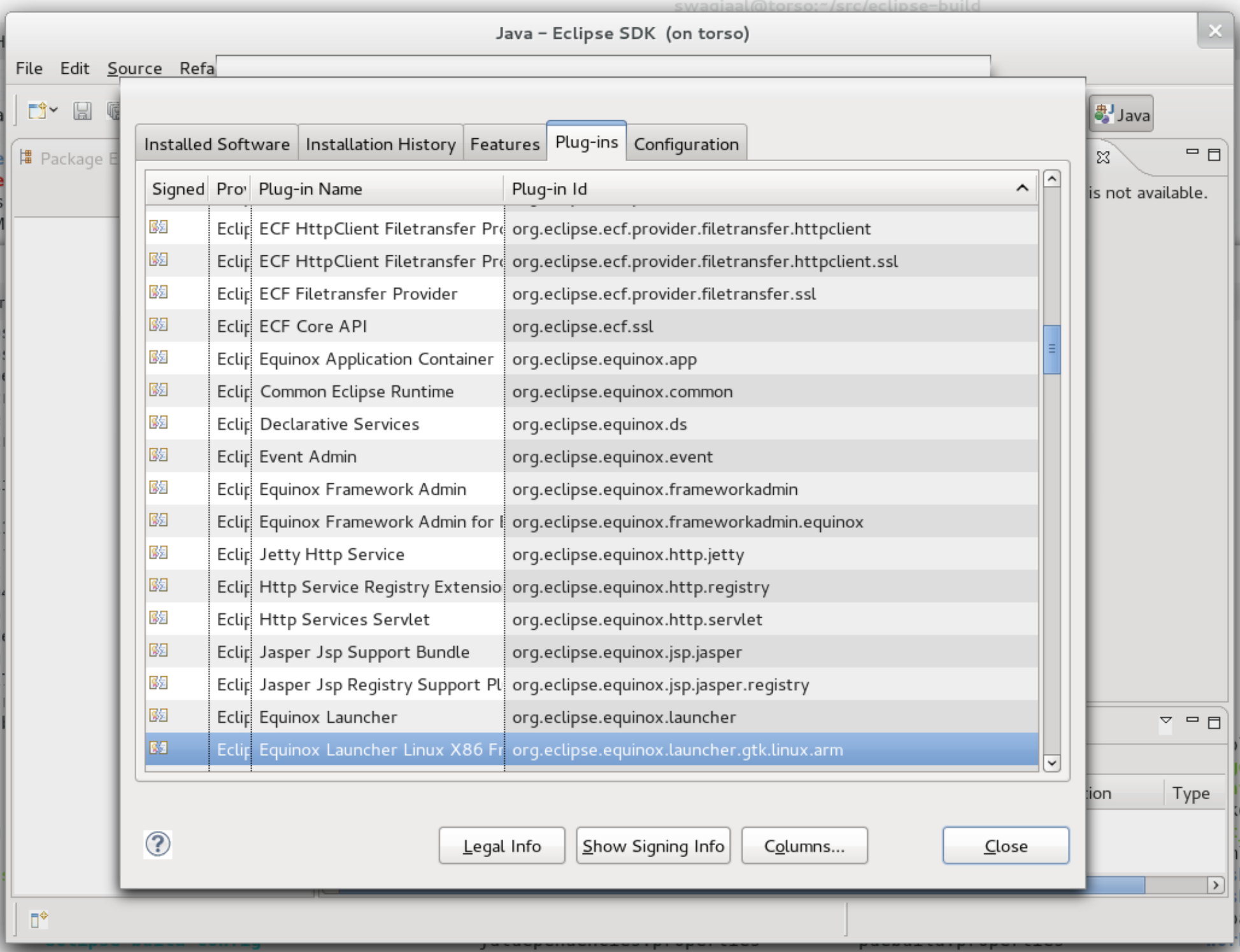
FPU

- Vector Floating Point (VFP)
- Podpora pro typy single a double
- SIMD operace (s vektory)
  - 8*single
  - 4*double

Mikroprocesory ARM v praxi

Mikroprocesory ARM v praxi

- Vývojové nástroje jako na x386/x86_64
  - C
  - C++
  - Fortran
  - Java
  - Crosskompilace i vývoj přímo na ARMu
- Podpora OS
  - Linux (Fedora!)
  - Windows 8???


Java – Eclipse SDK (on torso)
File Edit Source Refa
Java
Installed Software
Installation History
Features
Plug-ins
Configuration
Signed
Pro
Plug-in Name
Plug-in Id
Eclip ECF HttpClient Filetransfer Pr org.eclipse.ecf.provider.filetransfer.httpclient
Eclip ECF HttpClient Filetransfer Pr org.eclipse.ecf.provider.filetransfer.httpclient.ssl
Eclip ECF Filetransfer Provider org.eclipse.ecf.provider.filetransfer.ssl
Eclip ECF Core API org.eclipse.ecf.ssl
Eclip Equinox Application Container org.eclipse.equinox.app
Eclip Common Eclipse Runtime org.eclipse.equinox.common
Eclip Declarative Services org.eclipse.equinox.ds
Eclip Event Admin org.eclipse.equinox.event
Eclip Equinox Framework Admin org.eclipse.equinox.frameworkadmin
Eclip Equinox Framework Admin for org.eclipse.equinox.frameworkadmin.equinox
Eclip Jetty Http Service org.eclipse.equinox.http.jetty
Eclip Http Service Registry Extensio org.eclipse.equinox.http.registry
Eclip Http Services Servlet org.eclipse.equinox.http.servlet
Eclip Jasper Jsp Support Bundle org.eclipse.equinox.jsp.jasper
Eclip Jasper Jsp Registry Support Pl org.eclipse.equinox.jsp.jasper.registry
Eclip Equinox Launcher org.eclipse.equinox.launcher
Eclip Equinox Launcher Linux X86 Fr org.eclipse.equinox.launcher.gtk.linux.arm
Legal Info
Show Signing Info
Columns...
Close
is not available.
Type
eclipse-build-feature
eclipse-build-generatedScripts.tar.bz2
jdtnonosgidependencies.properties
junitHelper.xml
pdebuild.xml
provision.sdk-stamp

Technologie založené na ARMu

- Denver
  - Výkonný procesor zaměřený na desktopy a servery
- OMAP
  - Texas Instruments
  - Jádro ARM + další koprocesory
  - SoC
    - Všechny důležité obvody počítače na jednom čipu

Technologie založené na ARMu

- NVidia Tegra 3
  - Quad core
  - 1,4 Ghz
  - L1 Cache
    - 32kB I
    - 32kB D
    - Pro každé jádro
  - L2 Cache
    - 1MB
  - GPU
    - 3 jádra
    - ULP GeForce

Asus EEE Pad Transformer

Kapesní herní konzole

Beagle Board

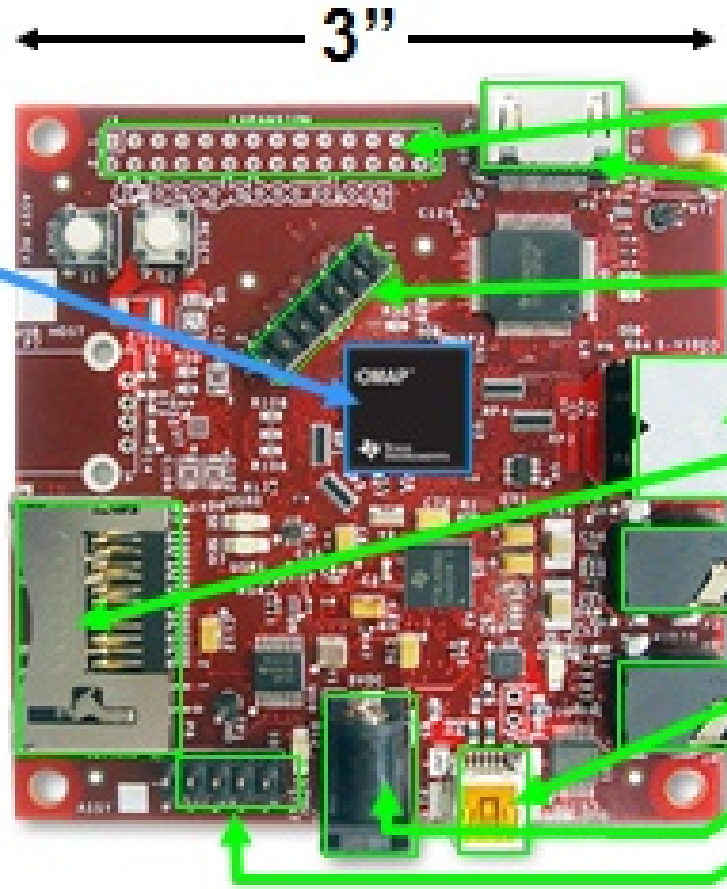
Laptop-like performance

TI OMAP3530

- 600 MHz superscalerARM® Cortex™-A8
- More than 1200 Dhrystone MIPS
- Up to 10 Million polygons per sec graphics
- HD video capable C64x+™ DSP core

Memory

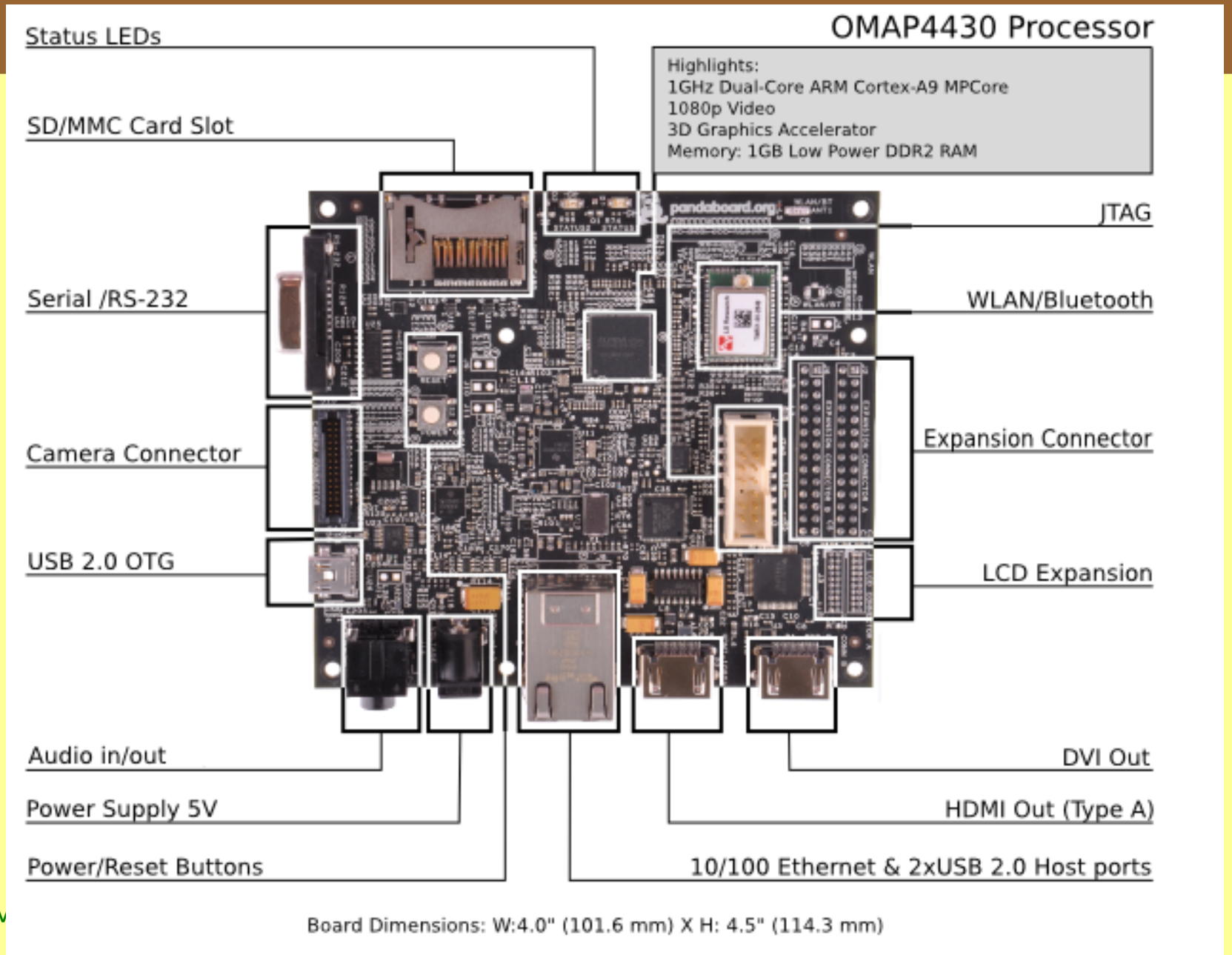
- 128MB LPDDR RAM
- 256MB NAND flash



Flexible expansion

- I²C, I²S, SPI, MMC/SD
- DVI-D
- JTAG
- S-Video
- SD/MMC+
- Stereo Out
- Stereo In
- USB 2.0 HS OTG
- Alternate Power
- RS-232 Serial

Panda Board


OMAP4430 Processor
Highlights:
1GHz Dual-Core ARM Cortex-A9 MPCore
1080p Video
3D Graphics Accelerator
Memory: 1GB Low Power DDR2 RAM
Status LEDs
SD/MMC Card Slot
Serial /RS-232
Camera Connector
USB 2.0 OTG
Audio in/out
Power Supply 5V
Power/Reset Buttons
JTAG
WLAN/Bluetooth
Expansion Connector
LCD Expansion
DVI Out
HDMI Out (Type A)
10/100 Ethernet & 2xUSB 2.0 Host ports
Board Dimensions: W:4.0" (101.6 mm) X H: 4.5" (114.3 mm)

Raspberry Pi

Na druhém konci výkonnostního spektra ...