

FreeBSD do zabudowy

Czyli nie tylko pecety

Rafał Jaworowski
raj@semihalf.com

meetBSD 2007, Warszawa



Szkic prezentacji

- Wprowadzenie do systemów wbudowanych (embedded systems)
- Przykłady
- Omówienie sytuacji FreeBSD w świecie embedded, potencjału z jednej strony, braków z drugiej
- Spojrzenie makroskopowe, bez szczegółów na poziomie kodu źródłowego lub konstrukcji sprzętu

Wprowadzenie

- Co to jest system wbudowany?
 - System mikroprocesorowy
 - CPU, pamięć, urządzenia I/O
 - Element większej całości
 - Na ogół wykonuje jedną specjalizowaną aplikację (zestaw aplikacji)
 - Silna integracja, małe rozmiary fizyczne, niski pobór energii
 - Nacisk na niezawodność, bezobsługowość, trwałość (odporność mechaniczną)
 - Brak GUI

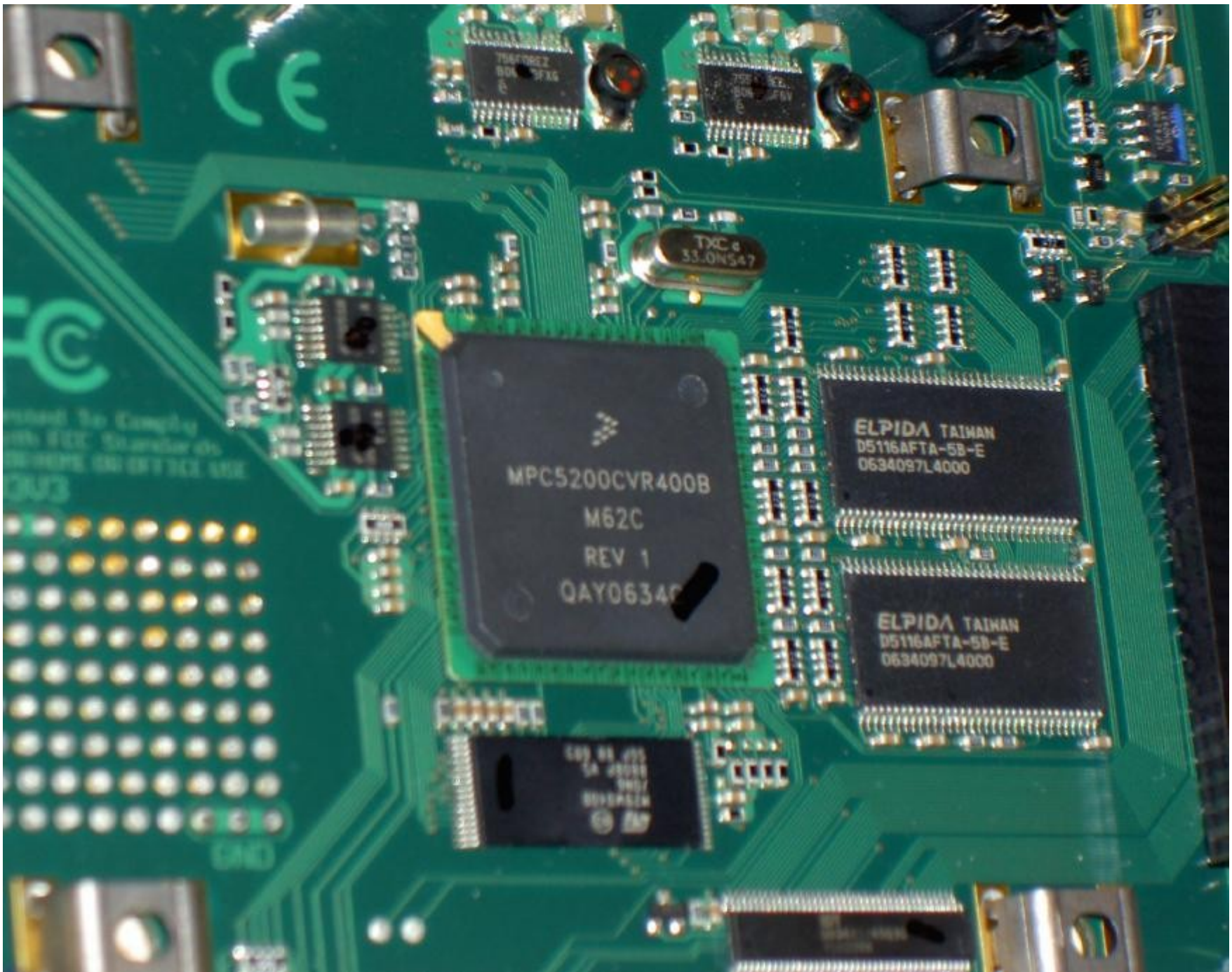
Wprowadzenie – c.d.

- Przykłady, zastosowania
 - Telefony, pralki, kuchenki, TV itp.
 - Auta, samoloty
 - Aparaty foto, kamery
 - Gry (konsole), PDA, odtwarzacze MP3
 - Sprzęt sieciowy (routery, firewalle)
 - Aparatura medyczna
 - Urządzenia wojskowe, kosmiczne
 - Mnóstwo innych...

Zanim pojawi się właściwy produkt

- Prototyp
 - Praca w warunkach laboratoryjnych
- Systemy uruchomieniowe
 - Przeznaczone do eksperymentów
 - Uniwersalna płytka drukowana
 - Łatwy dostęp do I/O
 - Dodatkowe układy programowanej logiki
 - JTAG

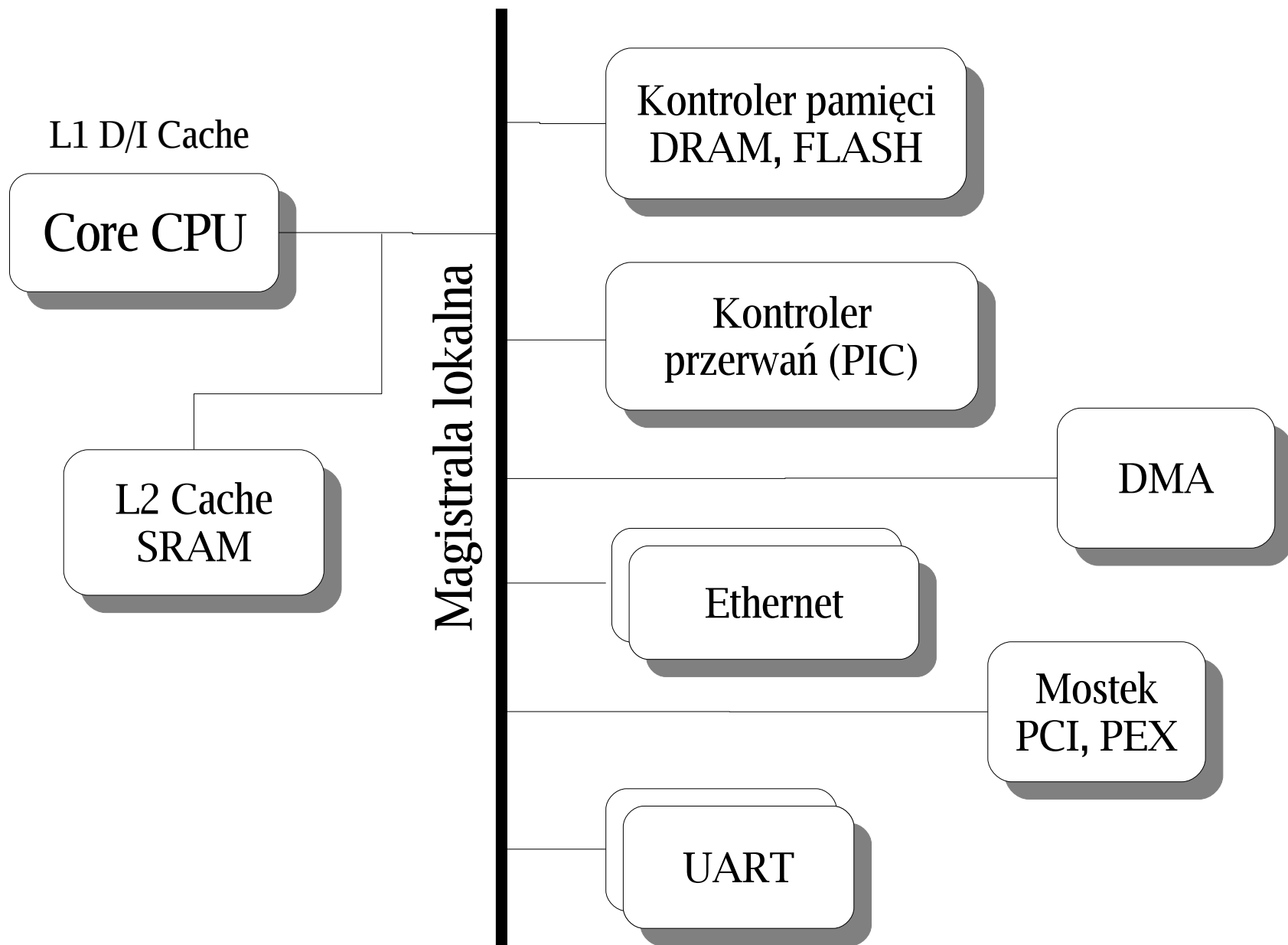




Terminologia

- Platforma
 - Kompletny system
 - Zestaw laboratoryjny – analogiczny do płyty głównej (potrzebuje jedynie zasilacza do działania)
- SoC (System-On-a-Chip)
 - Silnie zintegrowany układ
 - Lokalne magistrale
 - Kontroler(y) pamięci zewnętrznych (DRAM, FLASH etc.)
 - Układy peryferyjne (Ethernet, UART, mostki PCI, PEX, USB, DMA)
- CPU
 - Procesor główny
 - Część SoC

System-On-a-Chip



Systemy wbudowane

- Platforma
 - SoC + konfekcja (RAM, FLASH, PHY, obwody zasilania, resetowania itd.)
 - Specjalizowana logika (FPGA, CPLD)
 - Coraz więcej zasobów (MHz, pamięci)
- SoC
 - Specjalizowane bloki/moduły: crypto, telecom, pattern matching
- CPU (rdzeń)
 - 64-bit?
 - Multi-core, SMP?
 - Mogą być syntezywane w układach programowalnych!
- Przykładowe architektury
 - ARM, MIPS, PowerPC
 - RISC, super skalarne

O co w ogóle chodzi?

- Kogo interesują jakieś “małe” komputery..
- Ogromny rynek i potencjał
 - Setki milionów sztuk rocznie
- Są przecież już inne systemy...
 - Linux, NetBSD
- ...ale my lubimy FreeBSD!
- Zabawa czy na poważnie
 - Komercyjne produkty
 - R&D

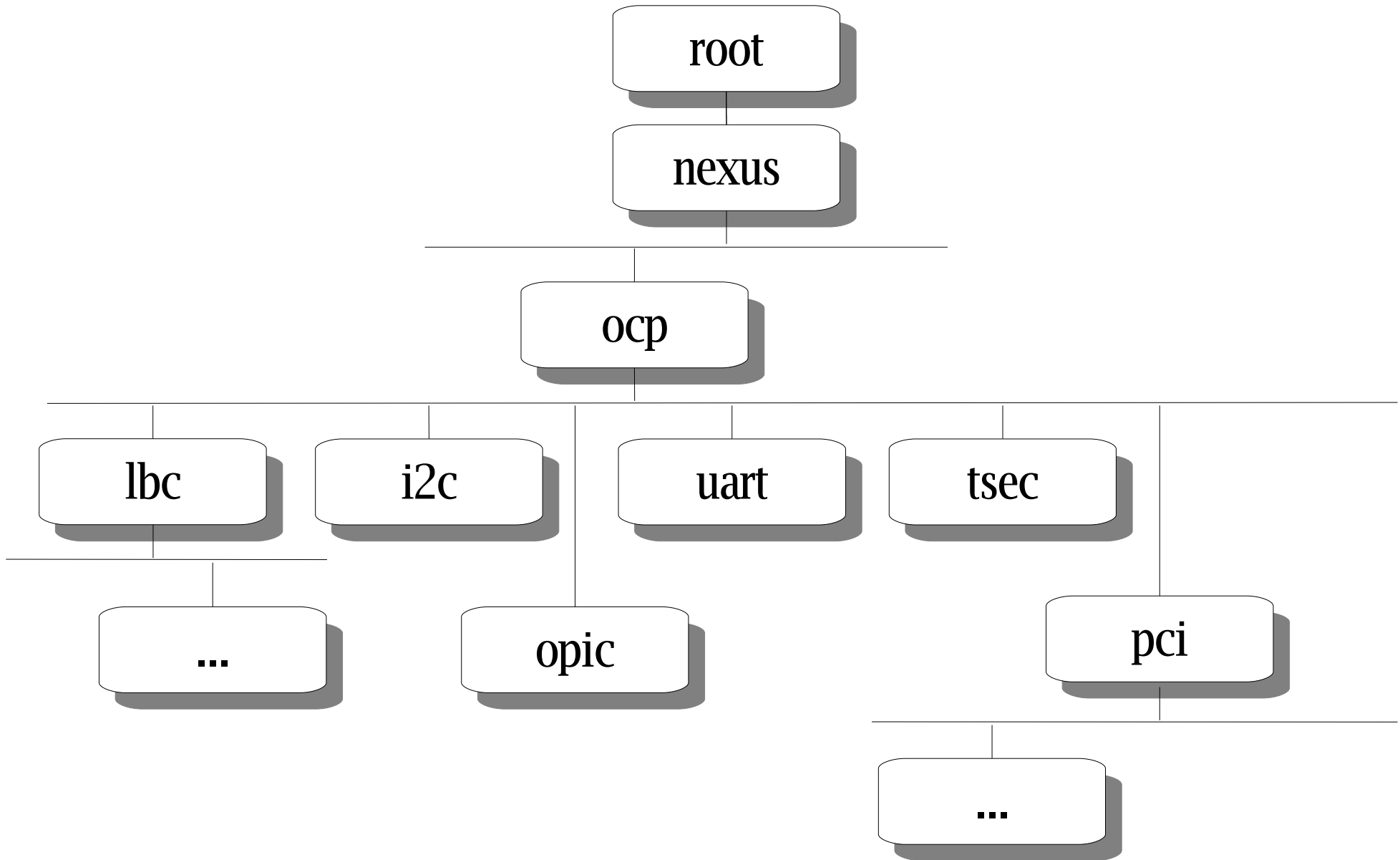
Port dla rodziny Freescale MPC85xx

- Platforma, zakres prac
 - Core e500, architektura zgodna z definicją Book-E PowerPC
 - Specjalizowany kontroler telekomunikacyjny PowerQUICC III
 - Kompletny port: loader, kernel + świat, wsparcie (sterowniki) dla głównych modułów SoC
- Najważniejsze zagadnienia
 - Inny model MMU
 - Wyłącznie TLB, brak segmentów, BATów
 - Dwupoziomowa tablica translacji (forward page table)
 - W efekcie: nowy moduł *pmap*

FreeBSD/MPC85xx c.d.

- Najważniejsze zagadnienia c.d.
 - Rozszerzony model wyjątków
 - Brak FPU – soft float
 - Sterowniki do zintegrowanych urządzeń
 - On-chip peripherals (Ethernet, PCI)
 - Większość sterowników napisana od zera
 - Pełne środowisko uruchomieniowe
 - rozszerzenie U-Boot – nowe API dla samodzielnych aplikacji
 - loader(8) – uruchamiany jako bootloader ostatniego poziomu
- Wspierane warianty
 - MPC8555, MPC8541, MPC8548

Model MPC85xx - *newbus*



Bliski horyzont, daleki plan

- Integracja wsparcia 85xx z drzewem FreeBSD
 - //depot/projects/e500/...
 - 8-current, backport 7.x
- Kolejne chipy Freescale, AMCC 4xx, inne (ARM)
- Kiedy FreeBSD na Playstation3?
 - Dwurdzeniowy POWER (zblizony do G5) jako CPU
 - 32-bit >> 64-bit
 - Brak *bridge mode*
 - Obecny port FreeBSD/powerpc jest wyłącznie 32-bitowy
 - Rozszerzony model MMU
 - *pmap* (64-bit PTE itd.)
 - SMP
 - loader(8) jako aplikacja hypervisor'a

Przenośność FreeBSD

- Środowisko
- Boot
 - loader(8) – sys/boot
 - libstand(3) – lib/libstand
- locore
 - sys/arm/arm/locore.S
 - sys/powerpc/booke/locore.S
 - mi_startup()
- bus_space(9)
- bus_dma(9)
- pmap(9)

Przenośność FreeBSD c.d.

– Plusy

- loader(8)
- Separacja części MD (machine dependent), MI (machine independent)
- Abstrakcja dostępu do urządzeń I/O, DMA
- Klarowny i kontrolowalny startup jądra
- Newbus
 - obiektowość

– Minusy

- Build system
 - Nie całkiem skrośny
 - Brak separacji toolchain'u
- UFS
 - Brak wsparcia cross-endian

FreeBSD w świecie embedded

- Obecne wsparcie dla ARM, MIPS, PowerPC
- Najwięcej platform i wariantów: ARM
 - Stosunkowo łatwo osiągalny hardware, zestawy uruchomieniowe, debuggery sprzętowe
- MIPS – Perforce
- PowerPC e500 – Perforce
- Ogólnie: skromny zakres wspieranych platform (w porównaniu z innymi systemami)
- Real time?
 - Interrupt latency
 - RT scheduling

Podsumowanie prezentacji

- Wstęp do systemów wbudowanych
- Omówienie terminologii
- Przykłady końcowych produktów
- Przykłady systemów laboratoryjnych
- Port FreeBSD/MPC85xx
- Plany
- Uwagi o przenośności środowiska FreeBSD na nowe architektury i platformy
- Ogólna kondycja FreeBSD w świecie wbudowanym

FreeBSD do zabudowy

Czyli nie tylko pecety

Rafał Jaworowski
raj@semihalf.com

http://www.semihalf.com/pub/meetbsd/2007_freebsd-do-zabudowy.pdf

meetBSD 2007, Warszawa

