Embedded Linux on DaVinci/OMAP Platforms

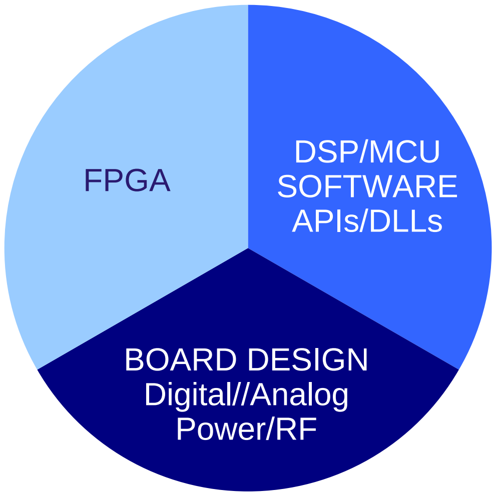
Presenters:
Tim Hiller
Jerry Johns

NUVATION

Texas Instruments

NUVATION designs embedded systems for technology organizations.

- Established since 1997
- 600+ successful projects completed
- Recognized world-wide as a leading Electronic Design Services firm
- Start-ups to Fortune-50 clientele, many clients in the area & globally
- Exceptional Engineering talent
- Proven Design Methodologies
- Fast-paced & fun corporate culture, execution-driven
- Unparalleled track record for predictable, repeatable results



Texas Instruments

Nuvation - TI

CLIENT, VENDOR & PARTNER

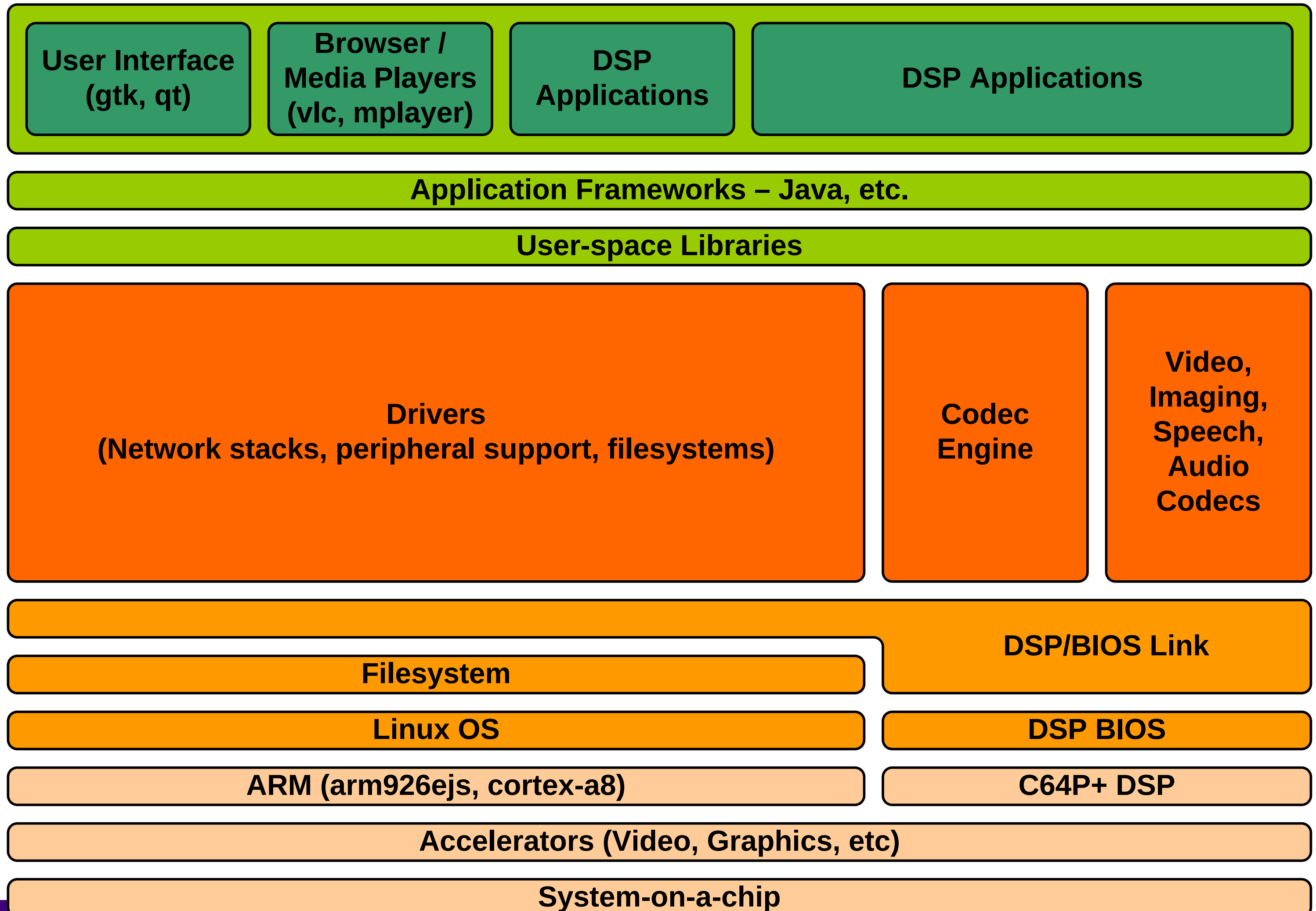
NUVATION

- TI DSP Third Party Network
- DaVinci Ecosystems Partner
- Authorized Software Reseller
- Low-Power RF Partner
- AEC Ecosystem Partner
- TI “Elite” Design House Partner
- Established vendor to TI
- DSP Champs 2006 Keynote
- TIDC keynotes by 3 Nuvation clients
- DSP Champs track on design methodologies
- TIDC Executive Tracks
- TIDC Exhibitor
- Vendor to TI Catalog DSP, HPA, DLP
- TI-based reference designs
- TI Tech Days Exhibitor & Presenter
- Texas Heat Battlebot creator

Why Linux?

- Open-Source!
- Cross-platform support
- Availability of documentation
- Developer Forums, Community support
- Driver development scales with complexity
- Extensive user-space APIs (threading, communications, driver access, etc.)
- Supports embedded scripting languages (Python, TCL, Lua)

Embedded System Stack-up



NUVATION

Texas Instruments

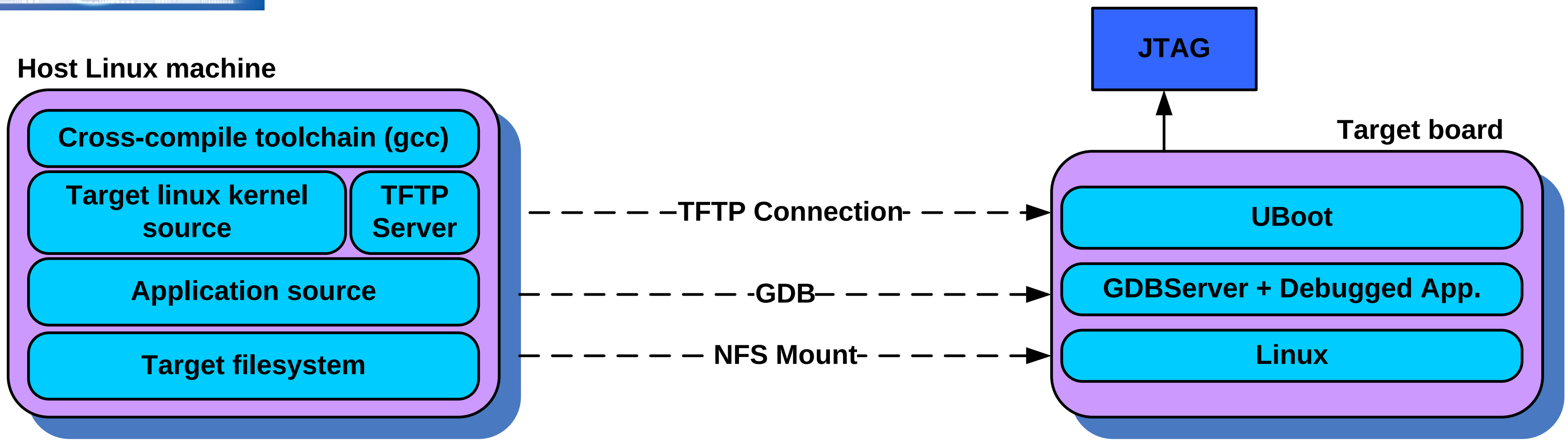
Typical Boot-Sequence

1st stage bootloader → 2nd stage bootloader → 3rd stage bootloader → Linux kernel boot → Filesystem mount → Application Execution

1st stage bootloader	2nd stage bootloader	3rd stage bootloader	Linux kernel boot	Filesystem mount	Application Execution
Read-only memory (ROM) loader	Davinci: UBL (universal boot loader OMAP: X-Loader	U-Boot		Flash, ATA, NFS, ramdisks	init.d, scripts

- 1st stage bootloader executes from read-only memory – contains mini drivers to load 2nd stage loader (through UART, NAND/NOR, etc.)
- 2nd stage bootloader loads into internal SRAM (limit of ~64KB)
- 3rd stage loads into external DDR2 memory (no size limit)
- Variety of booting methods in U-boot (TFTP, NAND/NOR, MMC)
- Linux usually re-initializes hardware during driver loading
- Filesystem contains application code as well as libraries, utilities

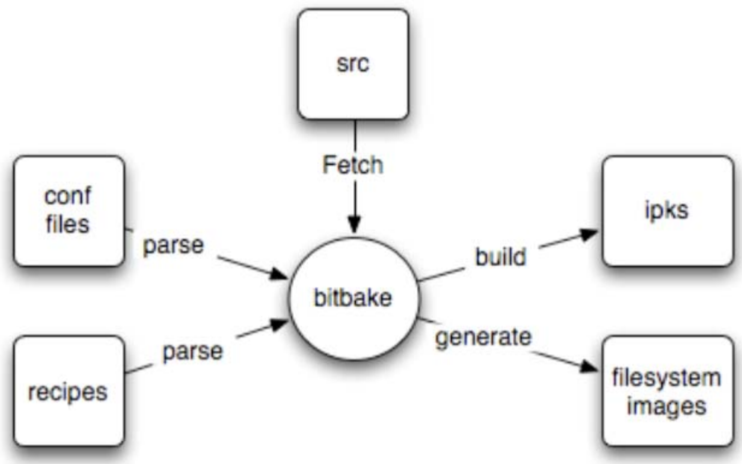
Standard Development Cycle

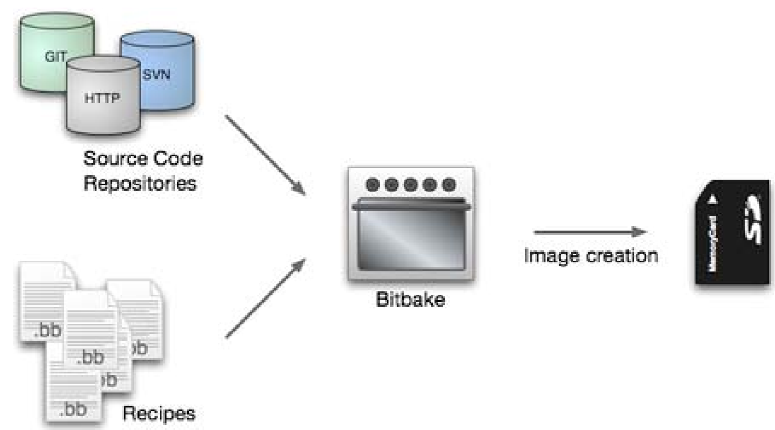


- Linux PC to host target toolchain and source code
- NFS Mounted filesystem to allow for rapid build-test-build cycles
- TFTP server to load kernel image through U-boot
- JTAG to flash NAND/NOR/OneNand with initial bootloaders
- GDB for debugging / single-stepping (integrated into Eclipse)

OpenEmbedded Development Flow

- Self-contained cross build system (used in OMAP3 platform)
- Currently used on Beagle Board, OMAP3EVM, Overo Gumstix
- Collection of 'recipes' that describe how to build bootloaders, libraries, applications
- Each recipe contains instructions on fetching source, unpacking, configuration, compilation and installation
- Does not include source code – fetches source from instructions in recipe
- Output is individual packages (ipks) and filesystem images





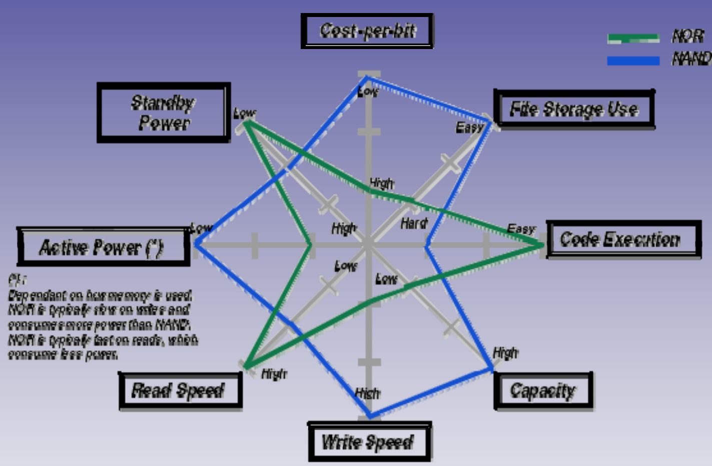
General Design Considerations

- Flash/Filesystem Selection
- Boot-time Optimizations
- Dealing with Memory
- Application Considerations
- Factory/Field Upgrades

Flash/Filesystem Selection

FLASH

- NAND vs. NOR vs. OneNAND



- DMA capabilities
- ECC Overhead
- OneNAND merges the best of both worlds (NAND Flash with NOR interface)

Filesystem

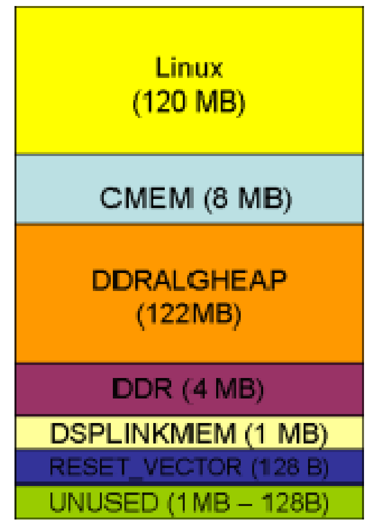
- Filesystem selection directly tied with choice of Flash device
- NAND **requires** wear-levelling support (for bad blocks) as well as ECC generation/correction
- JFFS2 common for NOR/NAND – log-structured, compressed storage
- All nodes must be scanned at run-time (slow)
- YAFFS2 is performance oriented - supports checkpointing for fast mount times
- UBIFS is successor to JFFS2 – supported on latest GIT kernel
- Ramdisks/initramfs – exec. in RAM

Boot-time Optimization

- **U-boot**
  - Use DWORD memcpy()
  - Use DMA for loading kernel image
  - Change bootdelay granularity to milli-seconds
  - Optimize asynchronous memory timings
  - Optimize CRC algorithm (existing version is byte-oriented)
- **Kernel**
  - Don’t have the image compressed
  - Deferred network interface setup
  - Remove un-needed drivers, convert to modules for deferred loading
  - Avoid bogoMIPS calculation (pass in as bootargs, lpj=xxx)
- **Filesystem**
  - Ramdisks/Initramfs
  - Prune scripts, minimize run-levels
  - Proper filesystem selection

DDR Memory Considerations

- **High-level planning:**
  - Linux, DSP, CMEM partitioning
- **Characterize usage**
  - Use Linux tools (vmstat, free, etc)
  - Codec memory usage profiling, e.g Engine_getUsedMem()
- **Application needs:**
  - Video frame buffers
  - Use shared memory (e.g. FIFO buffers)
  - Use memory-mapped I/O for speed-up and reduced memory footprint
  - No dynamic memory allocations in task-loops (external fragmentation)
  - Intermediate buffers in thread pipelines



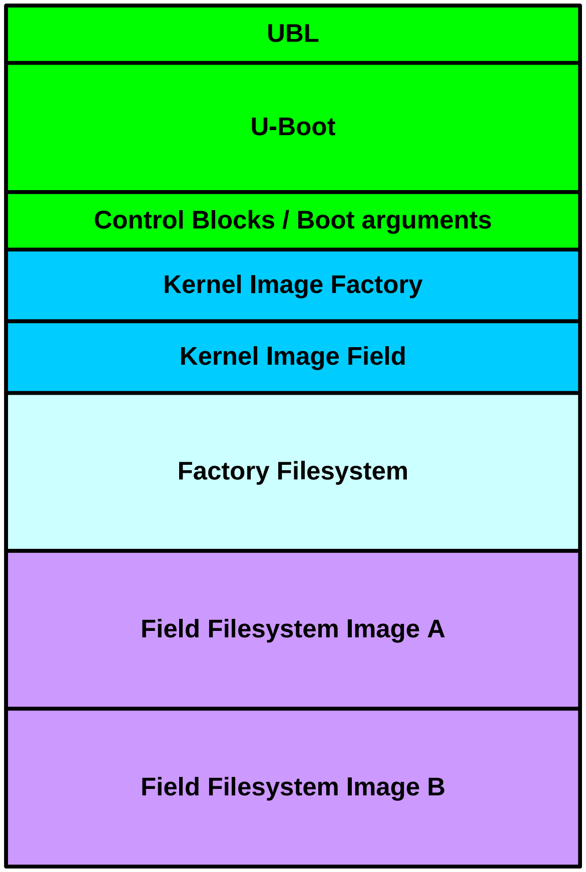
Sample Memory map

Application Considerations

- Use hardware peripherals when possible:
  - YUV422->YUV420 conversion (resizer block)
  - De-interlacing (DM6446 requires custom tap co-efficients)
  - Alpha-blending for video output (VPBE)
  - DMA for 2D sub-frame blitting
- DMA when transfer size > ~4K
- Up to 64-channels of DMA available, with chaining and 2D transfers
- Threads vs. Processes – latter uses more resources
- Compiler Optimizations (-O2 flag, -mfpu=neon for floating-point on OMAP3)
- Shared Libraries for reduced library memory footprint
- Application/Library stripping for release builds

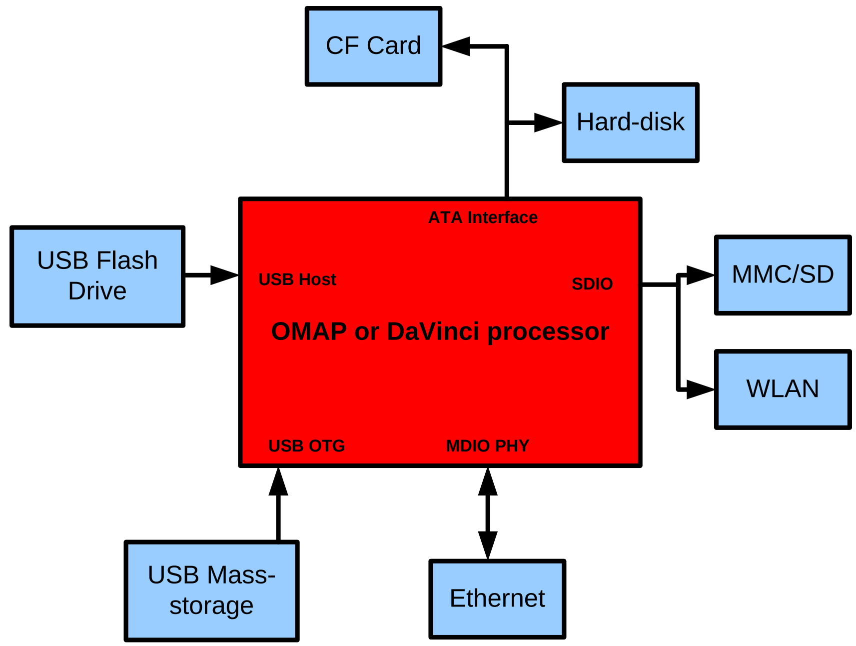
Factory Upgrading

- Need a seamless process to flash initial firmware on embedded platform
- OMAP/DaVinci ROM bootloaders have multiple peripheral boot options
- UART – Serial flashing toolkit available from TI
- USB – High Speed client interface
- Images need to be pre-packaged, e.g YAFFS2 has special ECC considerations
- Need factory partition (recovery)
- Sample partition map shown on right:



In-the-field upgrading

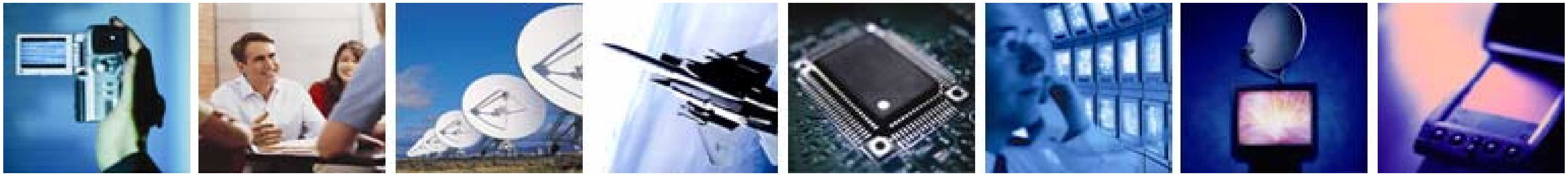
- Upgrading firmware in the field requires different focus
- Minimal customer intervention
- Usually done post-kernel boot to leverage extensive driver support
- Data transfer over USB, ATA, Network (Ethernet, WLAN)
- Safety mechanisms (failed boots lead to factory boot)
- Leveraging hardware to initiate safe-mode boot (push buttons)



NUVATION

SALES@NUVATION.COM
WWW.NUVATION.COM

Local Contact:
Mark Demers
Director, Sales & Business Development
Tel: 519.746.2304 x200
mark.demers@nuvation.com



WATERLOO DESIGN CENTER
470 WEBER STREET N, SUITE 201 WATERLOO, ON, CANADA N2L 6J2 TEL +1.519.746.2304 FAX +1.519.746.2939

SILICON VALLEY DESIGN CENTER
3590 NORTH FIRST STREET SAN JOSE, CA, USA 95134 TEL +1.408.228.5580 FAX +1.408.228.5590

Texas Instruments