Upstream in a Downstream Environment

Dinh Nguyen
Senior Embedded SW Engineer
Collaboration Summit 2016

ALTERA
now part of Intel

Introduction

- Embedded Linux Group at Altera(Programmable Solutions Group of Intel) Austin, TX
- Maintainer for arch/arm/mach-socfpga in Linux

Agenda

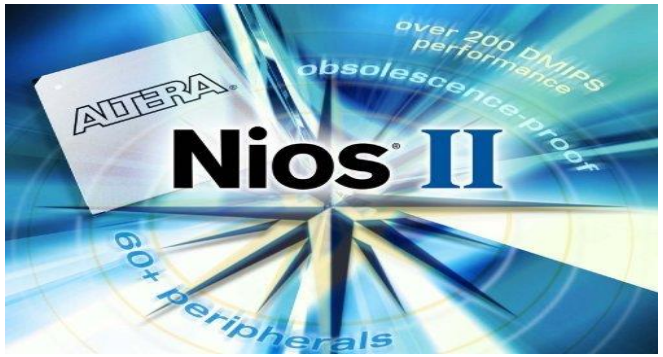
- Background of Altera's open source activity/non-activity
  - Focus on the Linux kernel and U-Boot
- Downstream environment and solutions
  - Network, machines, tools, legal framework
- Reap the Benefits Upstreaming
  - Upgrades, customers, suppliers
- Suffer the consequences of NOT Upstreaming
- Goal
  - Share war stories
  - Highlight how obstacles were overcame

Background: Altera

- Provides logic solutions which include FPGAs, SoCs, CPLDs and power management products.
  - FPGA = Field Programmable Gate Array
  - CPLD = Complex Programmable Logic device
  - → designed to be configured by a customer or a designer after manufacturing
  - SoC's combine ARM CPU's with FPGA's on the same die

Background

- What is SoCFPGA?
  - SoC + FPGA
  - ARM CPU + Hardened IPs + FPGA
  - Cyclone5/Arria5/Arria10
    - Dual Cortex A9 + FPGA
  - Stratix10 – Quad-core 64-bit (A53) + FPGA
  - Upstream effort started in 2012
- Nios II
  - Proprietary CPU architecture, designed to fit on Altera FPGAs
  - Upstream GCC support
  - Was completely down stream until v3.19
  - Now completely upstreamed



ALTERA
now part of Intel

Background: Altera's upstreaming activity

- In Linux kernel (kernel.org)
  - v3.5
    - 0 patches with altera.com emails
  - v4.5
    - 294 patches with altera.com emails
      - Support for SoCFPGA cyclone5, arria5 and arria10 devkits
      - Support for NIOS II
      - Drivers(USB, STMMAC, TSE, etc..)
    - 6 people from Altera listed in MAINTAINERS file
- In U-Boot
  - v2012.04
    - 0 patches with altera.com emails
  - V2016.03
    - 97 patches with altera.com emails
      - Support for SoCFPGA cyclone5, arria5

Background

What we knew about upstreaming!

- Management buy-in
- Upstreaming is important and will benefit.
  - See Greg KH's[1] many talks, and Tim Bird's[2] 2014 ELCE talk.
- Some patches could benefit competitor
  - Usually an issue for most companies
- Goal is to stay up to date with community release
- Scheduling
  - Upstream patches first
  - Cannot forecast patches acceptance
- No dedicated "upstreaming team"
  - Push comes to shove, upstreaming gets de-prioritized

Legal Environment

- Work with your legal department on a framework
  - Acceptance of upstreaming
    - What is upstreaming?
  - Which projects can you contribute to?
    - GPLv2, GPLv2+, BSD
  - What can/can't be upstreamed
    - Hopefully all kernel code can be upstreamed
  - Validity of confidential stamp on emails
    - Some corporations add legal disclaimers to emails

Differences between most Corporations and Community

	Most Corporations	Community
Email	Outlook	Evolution,Thunderbird, Pine, Mutt, text-based
Source Code Revision	ClearCase/CVS/SVN	GIT
Coding standards	Internal/Proprietary/Personal	/Documentation/CodingStyle
Issue Tracking	ClearQuest, FogBugz	Email, patchworks
Release strategy	Tarballs, ZIP files	GIT repo
Workstation	Microsoft Windows	Linux
IT security	Firewalls	Relatively Open

Environment(cont.)

- Email support
  - Avoid Outlook
    - Formatting issues
    - Cannot apply patches with ‘git am’
  - Company email policy
    - Legal disclaimers
  - Most have a SMTP port for outbound patches
  - Replies
    - Have an email address that you can access anywhere.
    - Use SMTP port on Evolution
      - Evolution has a work around to handle Outlook
  - Finally got opensource.altera.com

Environment(cont.)

- Source code management
  - Perforce/CVS/SVN/Clearcase
    - Hard to generate patches
  - GIT
    - Claim: “GIT is too hard!”
    - Workflow changes/benefits
    - Get GIT training!
- Coding standards
  - Internal vs. Community
- Issue Tracking
  - Added process
    - ClearQuest/FogBugz
    - Mixture of commit logs in GIT
  - Deploy Patchworks and pull in upstream patches

Environment(cont.)

- Release strategy
  - Tarballs/Package of files
    - Lose a lot of benefits from GIT
      - History of changes – ‘git blame’
      - Complete history of entire kernel
      - GIT bisect
  - Mixture of GIT repo and release package
    - Maintain GIT benefits
    - Deliver FPGA images
    - www.rocketboards.org/github
- Workstation
  - Virtual Machine
  - Dedicated Linux workstation
- IT Security
  - GIT protocol blocked
    - Separate network for open source work
    - Work remotely

Steps to Upstream a patch

- Get on latest kernel(master) or linux-next
  - Subsystem maintainer's own branch
- Develop/test
  - Build test allmoddefconfig/other architectures
  - Run checkpatch.pl
- Send patches via git send-email
  - Can get a lot of responses
  - Can get no responses
    - Friendly pings after ~a week or so
  - Can get a lot of bikeshedding[3]

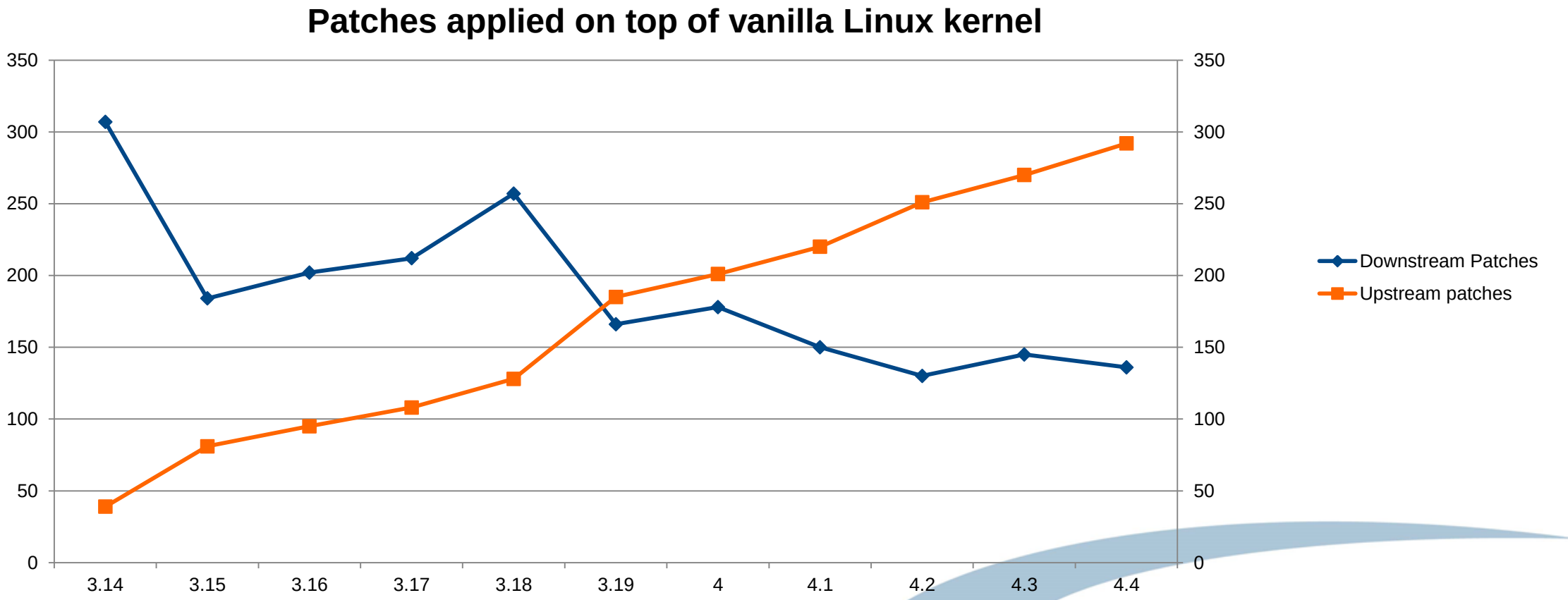
Steps to Upstream a patch(cont.)

- *"Continuous effort — not strength or intelligence — is the key to unlocking and using our potential." – Winston Churchill*
  - Important is to stay engaged with your patches
  - Convince maintainers why your patch(s) are important and should get merged

Handling patches

- Goal is to upstream patches first vs applying it locally first
  - Doesn’t really happen all of the time
    - Why?
      - Unfamiliar with upstream process
      - Management pressure to deliver
        - Classic mentality of “we can upstream it later” exists
      - Accountability
      - A matter of effort
        - Upstream patches are not a toss over the wall and forget
- Apply accepted upstream patches
    - DTS bindings do not have to change

Benefits of upstreaming



Benefits of Upstreaming(cont.)

- Linux upgrade
  - Simple as a ‘git rebase’ and fix a few conflicts
  - Take ~2 hours by 1 person
  - Can be handled by a small team(testing)
- Altera customers/partners feedback
  - 100% positive
  - Kernel updates can be done very quickly
  - Choices
- Testing
  - SoCFPGA Cyclone5 Devkit part of arm-soc board farm
  - Part of kernelci.org
  - Mainlined drivers get much more test coverage than any internal testing can cover
  - Constantly tested against many mainline

Consequences of not Upstreaming

- Different versions for different devices
  - Product cycles cannot keep up with Linux changes
  - 8.3 changes per hour in Linux v3.19 kernel [5]
  - v4.2: “1.09 million lines of code were added this time around with 285,000 removed, for a total growth of 800,000 lines of code.”[6]
  - Upgrades take more effort
- Cannot test against latest
  - No support for latest
- Effort to combine/upgrade?
  - Estimate is 2 – 4 weeks
  - Test effort doubles to triples
- Customers stuck on older versions
- Cherry-picking fixes extremely hard
- Community cannot help

Conclusion

- Obstacles can be overcome
- Enable the community!

References

- [1] https://www.youtube.com/watch?v=L2SED6sewRw
- [2] http://events.linuxfoundation.org/sites/events/files/slides/Overcoming_Obstacles_to_Mainlining-ELCE-2014-with-notes.pdf
- [3] https://en.wikipedia.org/wiki/Parkinson%27s_law_of_triviality
- [4] https://lkml.org/lkml/2015/8/13/545
- [5] https://www.youtube.com/watch?v=tE3804cOtXA
- [6] https://lwn.net/Articles/654633/

Thank You