AN AGILE APPROACH TO BUILDING RISC-V MICROPROCESSORS

- Authors : Berkeley RISC-V team
- Publication: IEEE Computer Society 2016

Key Contents:

- Agile Hardware Design Methodology
- RISC-V experience
- Chisel Language

Agile Manifesto

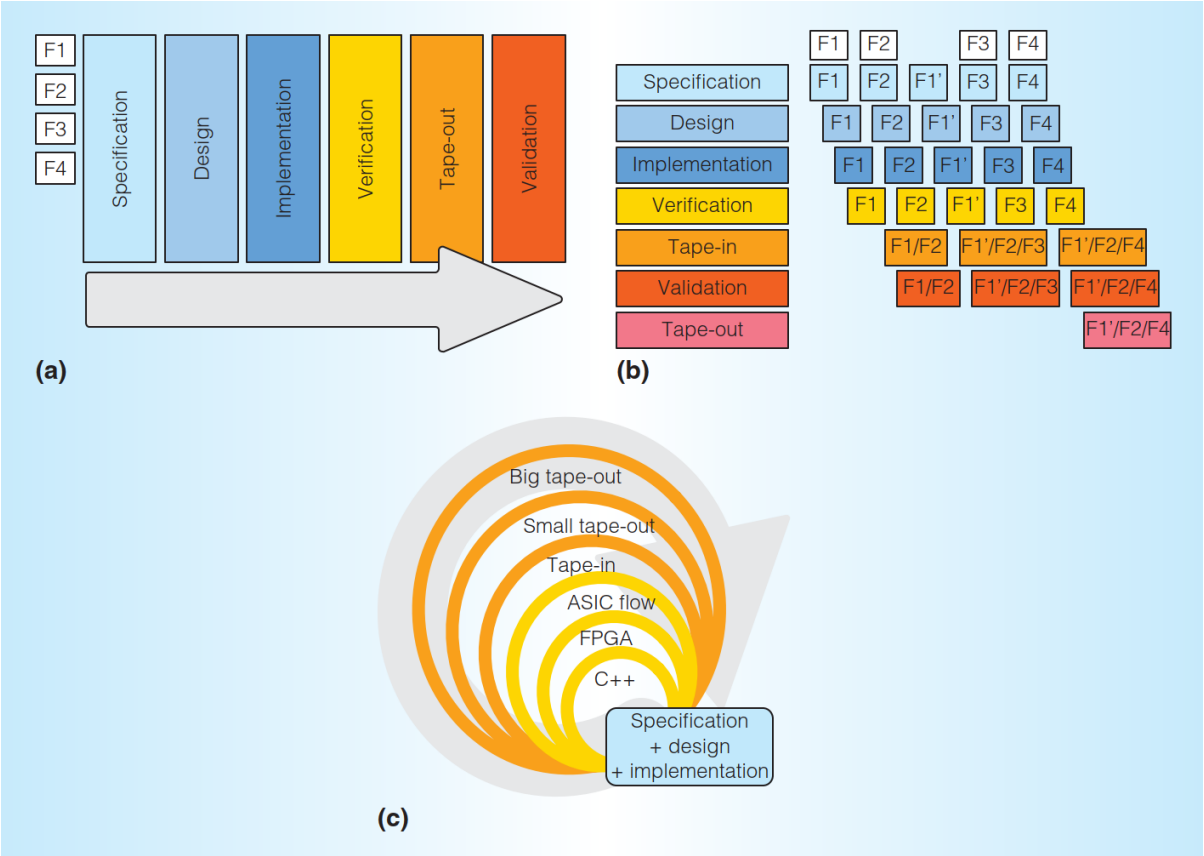
SoftWare Agile Manifesto

- 藉著親自並協助他人進行軟體開發，我們正致力於發掘更優良的軟體開發方法。透過這樣的努力，我們已建立以下價值觀:
  - 個人與互動 重於 流程與工具
  - 可用的軟體 重於 詳盡的文件
  - 與客戶合作 重於 合約協商
  - 回應變化 重於 遵循計劃
- 也就是說，雖然右側項目有其價值，但我們更重視左側項目。

Hardware Agile Manifesto

- **Prototyping**: Incomplete, fabricatable prototypes over fully featured models. (未完成，可替換的原型優於完整特性的模型)
- **Co-working**: Collaborative, flexible teams over rigid silos. (合作彈性的團堆優於剛硬的穀倉, 破除跨團隊的穀倉效應)
- **Tools**: Improvement of tools and generators over improvement of the instance. (改善工具與產生器優於實施例)
- **Agility**: Response to change over following a plan.(對改變的迅速反應優於詳實計畫)

Incomplete, fabricatable prototypes over fully featured models



- Typical Waterfall model: currently used in most design company.
  - waterfall approach is in *danger of grossly overshooting any tape-out deadline*
- Agile model: push minimum working model, adding features iteratively
  - will always have an available tape-out candidate with some subset of features for any tape-out deadline
  - validation using full rtl implementation.
  - guarantees cycle accuracy.
  - give early feed-back on back-end physical design issues.(QoR : cycle time, area, and power)

Collaborative, Flexible Teams over Rigid Silos

- Traditional hardware design teams usually are organized around the waterfall model.
  - communication costs:
    - Extensive effort to communicate design intent for each block
    - Rich documentation by the preceding stages
    - Mis-understanding between two function blocks.
- QoR not guarantees for whole flow.
- End-to-end iteration too expensive.
- Agile model:
  - engineers work in collaborative.
  - driving a feature for all implementation stage and,
  - avoiding most of documentation overhead required.
  - iterative from high-level architecture to low-level physical implementation.

Improvement of Tools and Generators over Improvement of the Instance

- Modern software relay on extreme reuse of extensive software library.
- Most commercial hardware design flows rely on rather **primitive languages**(aka verilog) to describe hardware.
  - hinder development of truly reusable hardware components
  - lead to a focus on building a single instance instead of designing components that can be easily reused in future designs.
- Agility HW design require better HDLs:
  - Agile Software emphasis collaborative better than specific tools.
  - But Modern HW is far less automated and far less reuse.
  - importance of reuse and automation enabled by new tools in the hardware design sphere.

Implementing the Agile Hardware Methodology

RISC-V

- free, open, and extensible ISA

Chisel

- Chisel is embedded in Scala
- Hardware developers can tap into Scala's modern
- object-oriented programming
- functional programming
- parameterized types
- abstract data types
- operator overloading
- type inference to improve designer productivity by raising the abstraction level and increasing code reuse

Comments

- Agile design methodology is suitable for developing for new/unstable standard.
  - Fast Prototyping, iterative approach to enrich features
  - Study from doing. Will do better for 2nd times.
  - Adaption to the change of the spec.
- Hardware Agile methodology requires cross-domain team and tools
  - An Agile Team need cross-domain team members: aka. SD,DD(front end/backend) and SE.
  - Adopting Chisel language cowork with C++ and FPGA for system and HW design verification.
  - Chisel language is good stuff! can simplify the coding effort on HW design.