Writing an operating system in 2.5 years

Yunhao Zhang

But first, writing an OS in 0.5 years

- P0: understand C and user-level instructions
- P1: understand context-switch and multi-threading
- P2: understand exception and privilege levels
- P3: understand file system abstractions
- P4: understand file system implementation
- P5: understand bus and I/O devices

https://github.com/yhzhang0128/egos-2000/blob/main/earth/sd/sd_rw.c#L13

Why 2.5 years? An overview



Obstacles

Summer 2020

Fall 2020





In June 2020, we created egos

- ~20K lines of code
- run on Intel/Arm CPU
- run as a Linux / MacOS process



20K lines of code

Students read a very small portion



2K lines of code

n Students read a large portion



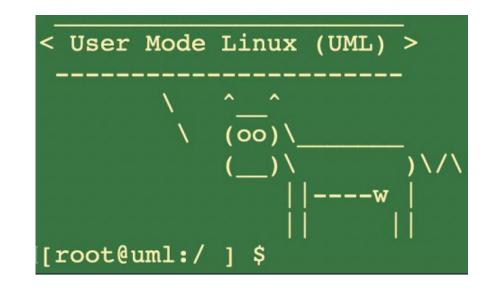
Intel x86 (1987)

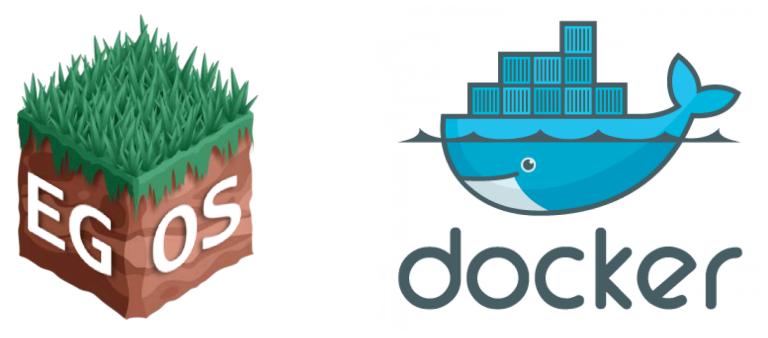
CPU document has several thousands of pages



RISC-V (2010)

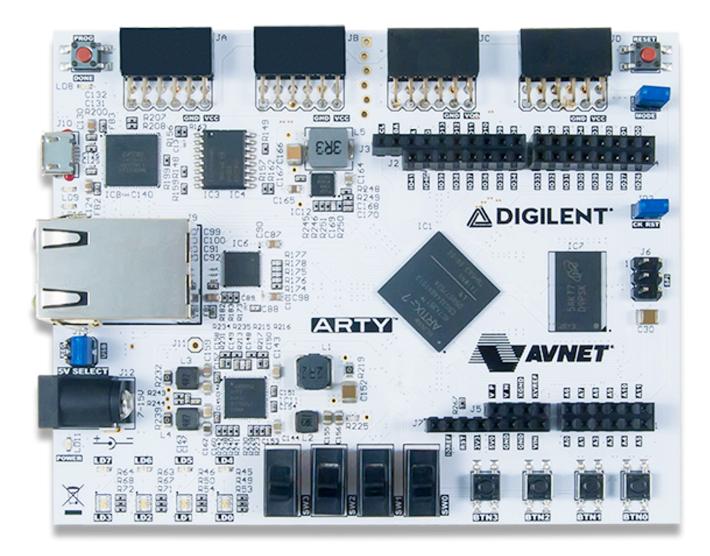
CPU document has <100 of pages





User-mode OS

Easier to deploy and run



OS on hardware More realistic and fun



The motivation

$\sim 20K \rightarrow \sim 2K$

$x86 \rightarrow RISC-V$

Linux / MacOS \rightarrow hardware

non-experts like students, my friend doing ML theory, etc.

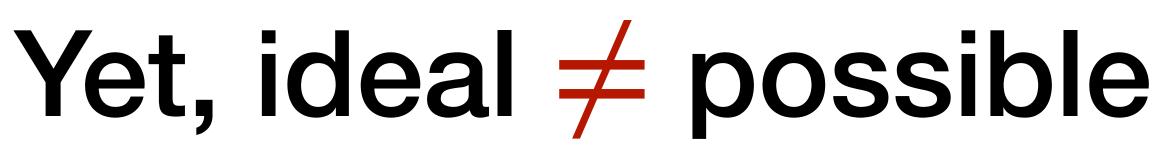
Lesson Good motivations should convince non-experts why the work is important



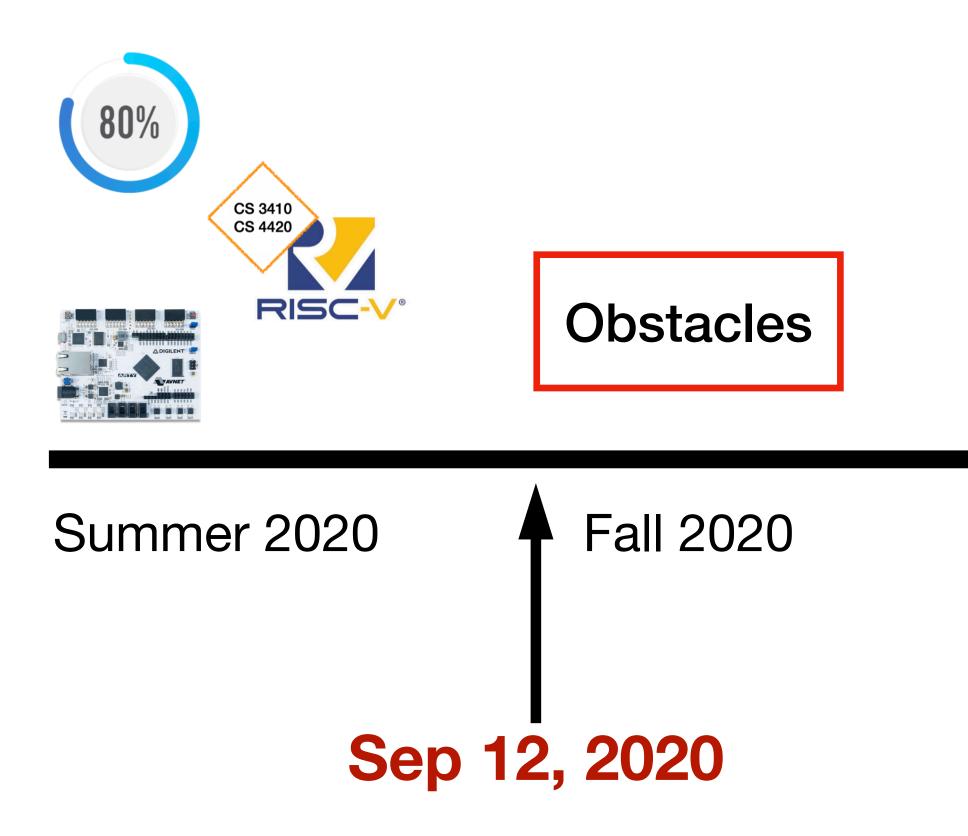


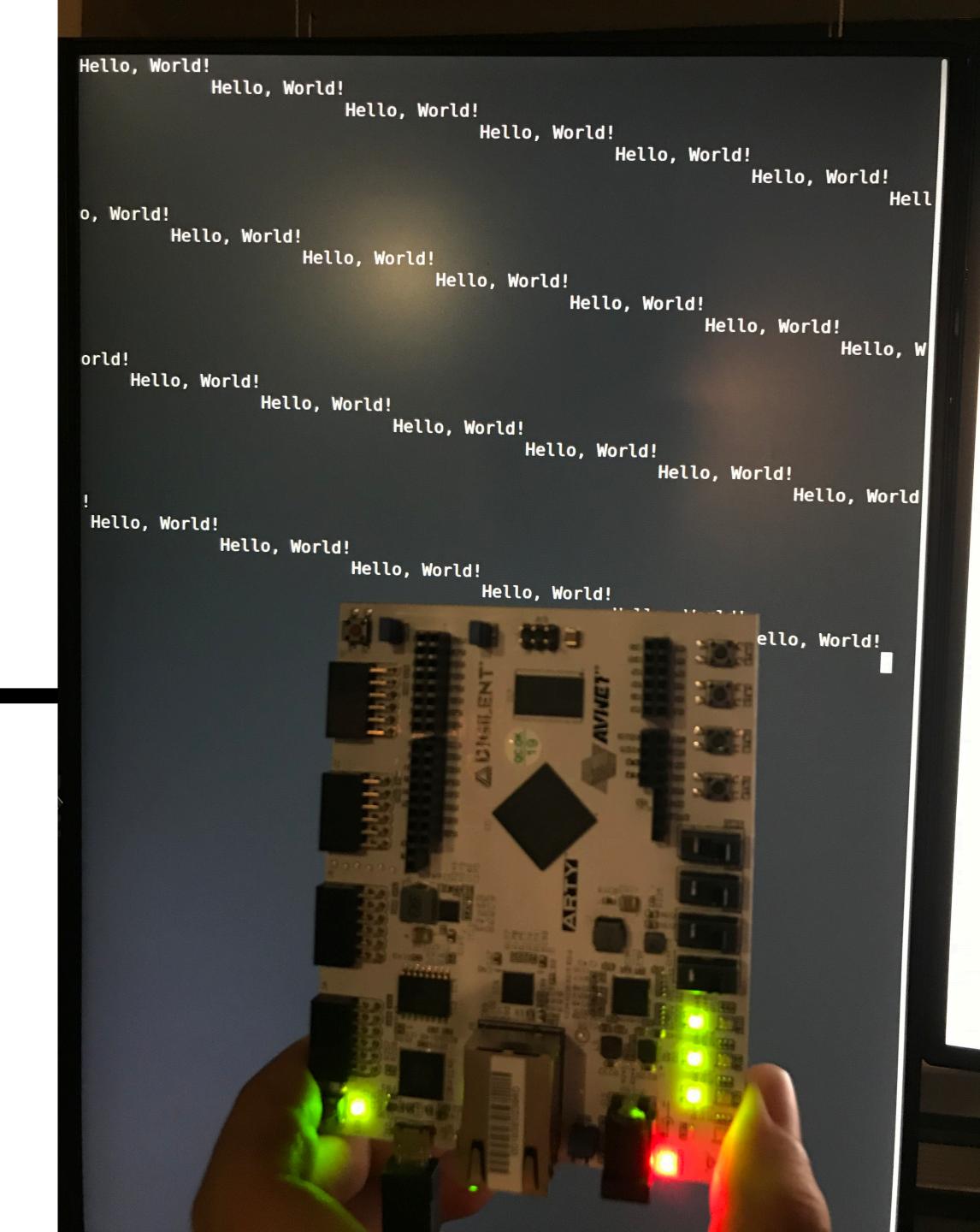
Summer 2020

Fall 2020



Hello World





Obstacles & Hope









The hardware supports privilege levels and exceptions

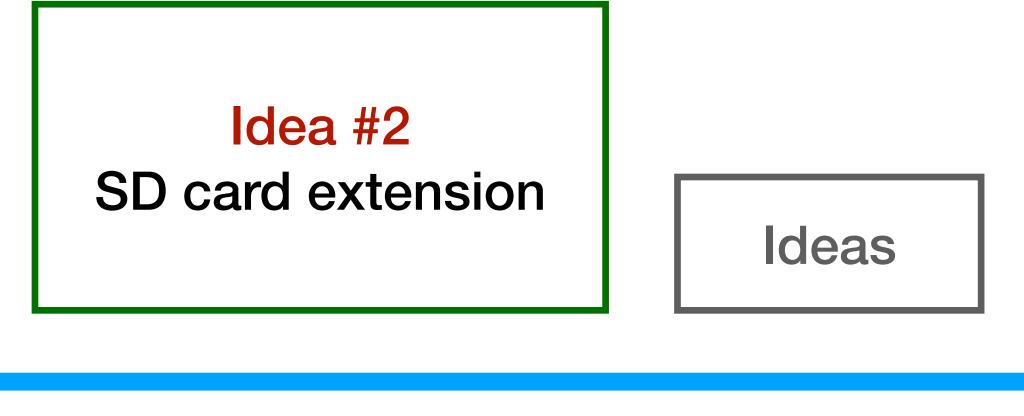
Overcome obstacles with ideas!

Idea #1 Increase memory

Obstacles

Fall 2020

Jan 2021



Nov 2021 Fall 2021

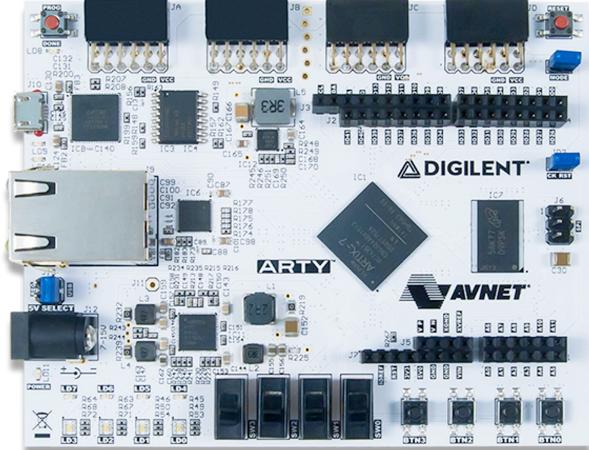
Open-source hardware

Search or jump to / Pulls Issues Codes	paces Marketplace Explore 🗘 + 🗸 🏤 🕇
This repository has been archived by the owner before	ore Nov 8, 2022. It is now read-only.
Sifive / freedom Public archive	187 ▼ 😵 Fork 269 ▼ 🟠 Star 1k ▼
<> Code Issues 64 Pull requests 6 Actions [🗄 Projects 🔃 Security 🗠 Insights
ਿੱ master - Go to file	<> Code - About
erikdanie Update README.md on Mar 1, 2	Source files for SiFive's Freedom platforms
bootrom sdboot: auto-extract tl-clock frequency	4 years ago
➡ fpga-shells @ 14 Bump fpga-shells, supports vc707 with	3 years ago ☆ 1k stars
nvidia-dla-blocks nvidia-dla-blocks: bump to point at a p	4 years ago 187 watching
rocket-chip @ b2 updated submodules	4 years ago % 269 forks
➡ sifive-blocks @ a updated submodules	4 years ago
src/main/scala Added BTB and a 16kB 2-way I-Cache	3 years ago Releases 1
 .gitignore Initial commit.	6 years ago S Freedom E300 Arty De Latest
.gitmodules Revert url of submodule for pull request	4 years ago
LICENSE Initial commit.	6 years ago Packages
Makefile.e300art rocket-chip: bump for API changes	5 years ago No packages published
Makefile.vc707-io vc707-iofpga: design runs at 200MHz	5 years ago

Running open-source hardware

Search or jump to	/ Pulls Issues	Codespaces Market	tplace Explore 🗘 + 🗸 🌆 🗸	
This rep	pository has been archived by the o	wner before Nov 8, 2022	2. It is now read-only.	
☐ sifive / freedom (Pu	Image: Sifive / freedom Public archive Image: Watch 187 → Image: Star 1k → Image: Sifive / freedom Image: Star 1k →			
<> Code 🕑 Issues 64	<> Code 🕢 Issues 64 II Pull requests 6 🕑 Actions 🖽 Projects 🛈 Security 🗠 Insights			
ੀ master →	Go	to file <> Code -	About	
erikdanie Update REA	ADME.md c	on Mar 1, 2021 🕚 154	Source files for SiFive's Freedom platforms	
bootrom	sdboot: auto-extract tl-clock freque	ncy 4 years ago	🛱 Readme	
🗗 fpga-shells @ 14	Bump fpga-shells, supports vc707 w	vith 3 years ago	화 Apache-2.0 license ☆ 1k stars	
nvidia-dla-blocks	nvidia-dla-blocks: bump to point at a	a p 4 years ago	 187 watching 	
🔄 rocket-chip @ b2	updated submodules	4 years ago	父 269 forks	
sifive-blocks @ a	updated submodules	4 years ago		
src/main/scala	Added BTB and a 16kB 2-way I-Cacl	ne 3 years ago	Releases 1	
🗋 .gitignore	Initial commit.	6 years ago	S Freedom E300 Arty De Latest	
] .gitmodules	Revert url of submodule for pull requ	uest 4 years ago		
	Initial commit.	6 years ago	Packages	
🗋 Makefile.e300art	rocket-chip: bump for API changes	5 years ago	No packages published	
Makefile.vc707-io	vc707-iofpga: design runs at 200MF	lz 5 years ago		

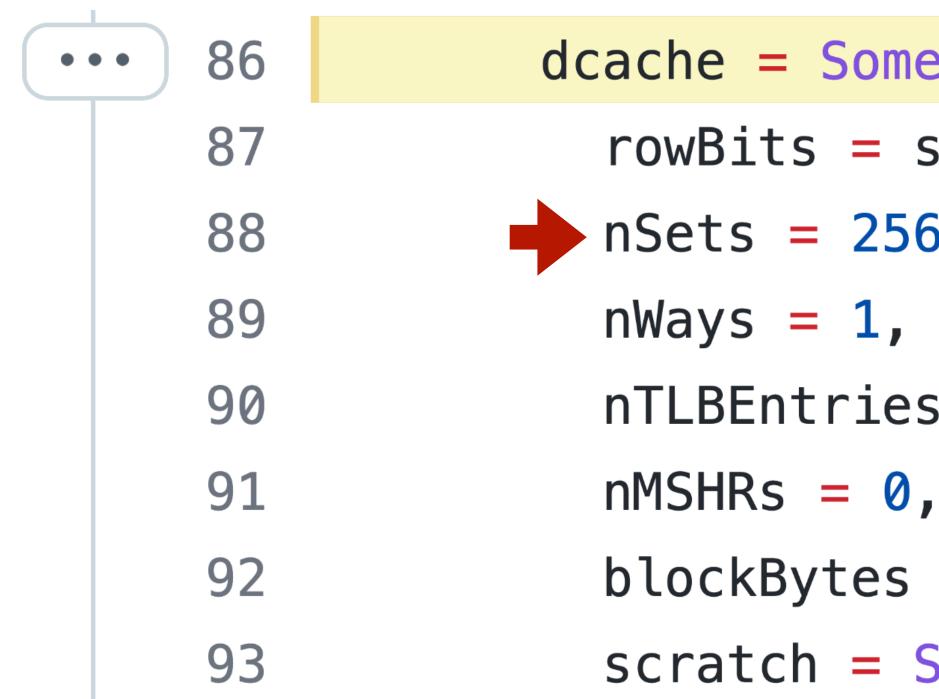
A binary file encoding the hardware design (clocks, registers, circuits, etc.)



FPGA: emulate the hardware design

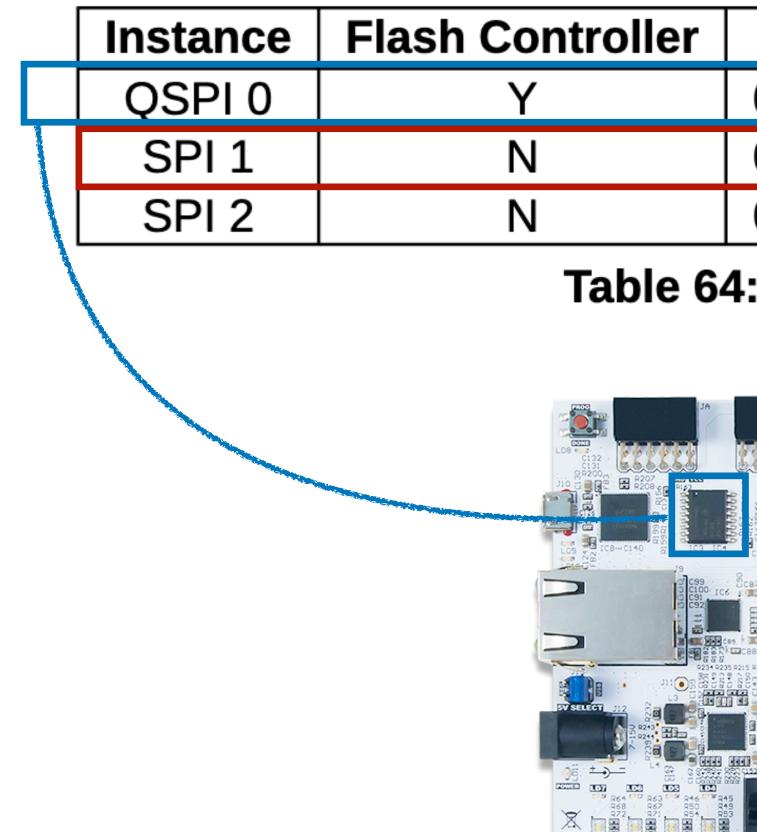


Idea #1: Increase memory



https://github.com/chipsalliance/rocket-chip/blob/ b21c7879b3ea22f69cb8457109561f37c225f8ea/src/main/scala/subsystem/Configs.scala#L78

- dcache = Some(DCacheParams()
 - rowBits = site(SystemBusKey).beatBits,
 - nSets = 256, // 16Kb scratchpad
 - nTLBEntries = 4,
 - blockBytes = site(CacheBlockBytes),
 - scratch = Some(0x8000000L))),

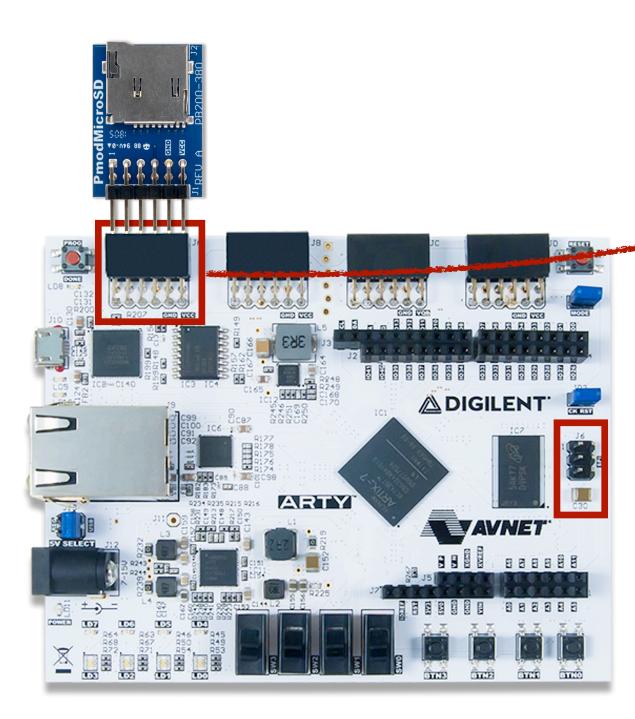


Chapter 19 of Sifive FE310 manual, v19p04 https://github.com/yhzhang0128/egos-2000/blob/main/references/sifive-fe310-v19p04.pdf

Background of the SPI bus

Address	cs_width	div_width	
0x10014000	1	12	
0x10024000	4	12	
0x10034000	1	12	

Instance	Flash Controller	Address	cs_width	div_width	
QSPI 0	Y	0x10014000	1	12	
SPI 1	N	0x10024000	4	12	
SPI 2	N	0x10034000	1	12	

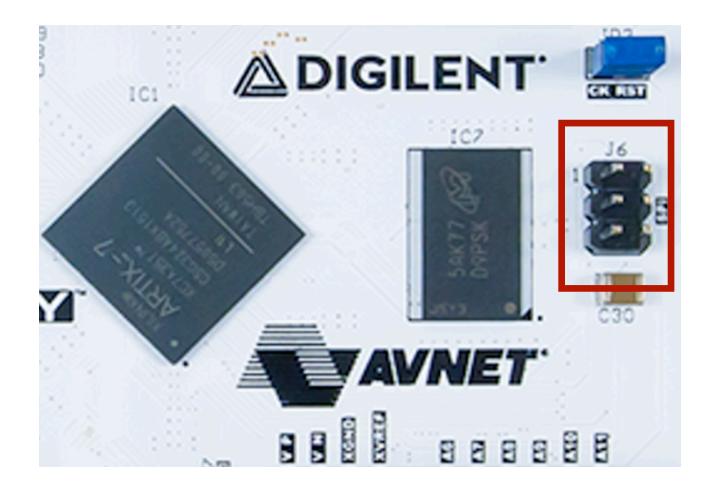


new SPI1

Remap SPI1 to microSD card

 Table 64:
 SPI Instances

old SPI1



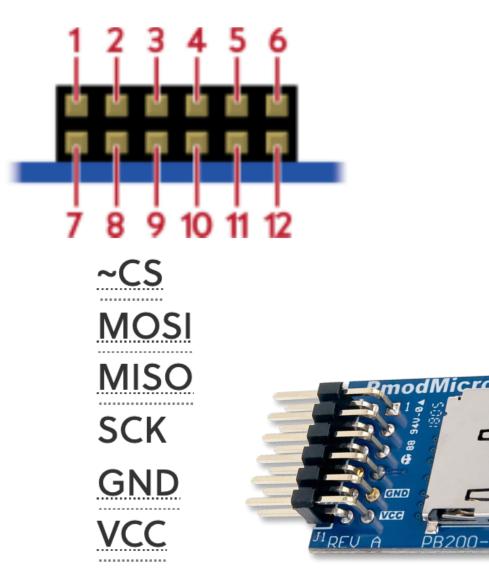


Old SPI1 mapping

Remap SPI1 to microSD card

6 pins GND + VCC + SPI(4)

Pin	1
Pin	2
Pin	3
Pin	4
Pin	5
Pin	6



New SPI1 mapping



Idea #2: Using microSD card as disk

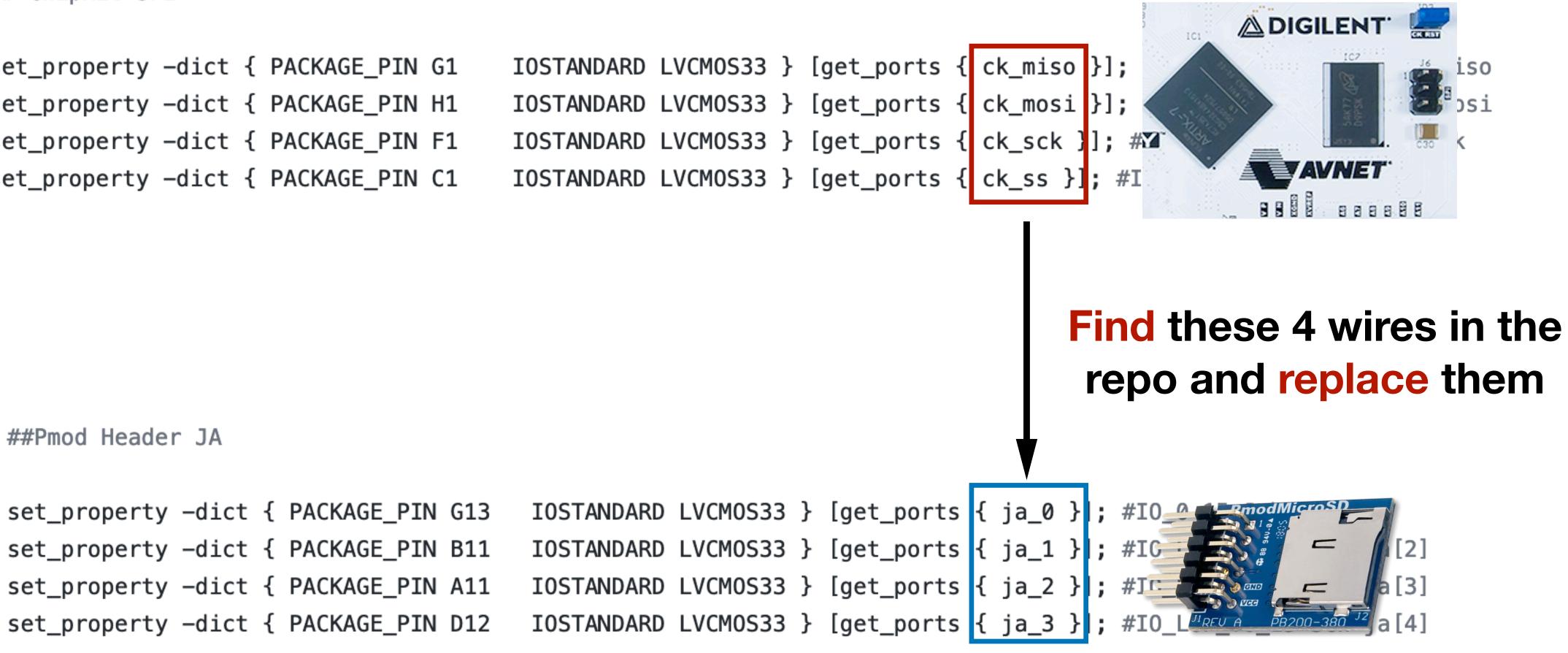
ChipKit SPI

- set_property -dict { PACKAGE_PIN G1
- set_property -dict { PACKAGE_PIN H1
- set_property -dict { PACKAGE_PIN F1
- set_property -dict { PACKAGE_PIN C1

```
##Pmod Header JA
```

set_property -dict { PACKAGE_PIN G13 set_property -dict { PACKAGE_PIN B11 set_property -dict { PACKAGE_PIN A11

https://github.com/sifive/fpga-shells/blob/14297af2878dc648ffd5751010fa72094ff444b0/xilinx/arty/constraints/arty-master.xdc#L48





The road to ideas is difficult

Obstacles

Fall 2020

- No concrete progress for >1 year
- Not sure whether this can succeed at all
 - Only person working on this project



Fall 2021

Take-away: Ideas are difficult to come up with and there is no guarantee of success





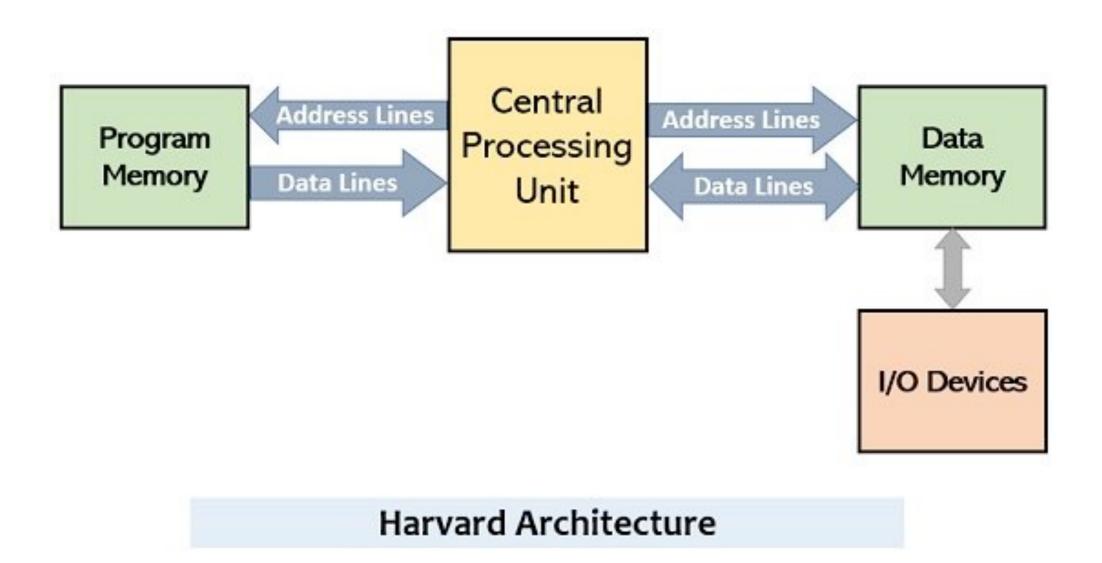
Obstacles

Summer 2020 Fall 2020

https://github.com/yhzhang0128/egos-2000/blob/main/references/README.md#software-development-history

The bug taking me >1 day to fix

```
core = RocketCoreParams(
 useVM = false,
 fpu = None,
 mulDiv = Some(MulDivParams(mulUnroll = 8))),
btb = None,
dcache = Some(DCacheParams(
 rowBits = site(SystemBusKey).beatBits,
 nSets = 256, // 16Kb scratchpad
 nWays = 1,
 nTLBEntries = 4,
 nMSHRs = 0,
 blockBytes = site(CacheBlockBytes),
 scratch = Some(0x8000000L))),
icache = Some(ICacheParams(
 rowBits = site(SystemBusKey).beatBits,
 nSets = 64,
 nWays = 1,
 nTLBEntries = 4,
  blockBytes = site(CacheBlockBytes)))))
```



https://github.com/chipsalliance/rocket-chip/blob/b21c7879b3ea22f69cb8457109561f37c225f8ea/src/main/scala/subsystem/Configs.scala#L78



Lesson Implementing a system is non-trivial and requires determination and hard work



Summer 2018

2 years: Becoming familiar with OS education

Then, challenge state-of-the-art

Motivation

Obstacles

Summer 2020

A 4.5-year research process

Summer 2020

Ideas

Implementation

Evaluation



Next step: Publish the research

Lines of Code	What?	Lines of Code	What?
199	Boot Loader & TTY Driver	336	File System
182	SD Card Driver	264	Applications & Daemons
32	Interrupt & Exception Handling	269	Library & Networking (TBA)
137	Page Table & Software Translation	64	Makefile
345	Timer, Scheduler & System Call	172	RISC–V Emulator & Board Tools

Fighting for a world where every college student can read all the code of an operating system