A few trips later...

I could open my own PSTN

- A total of 12 *additional* phones were rescued.

I could open my own PSTN

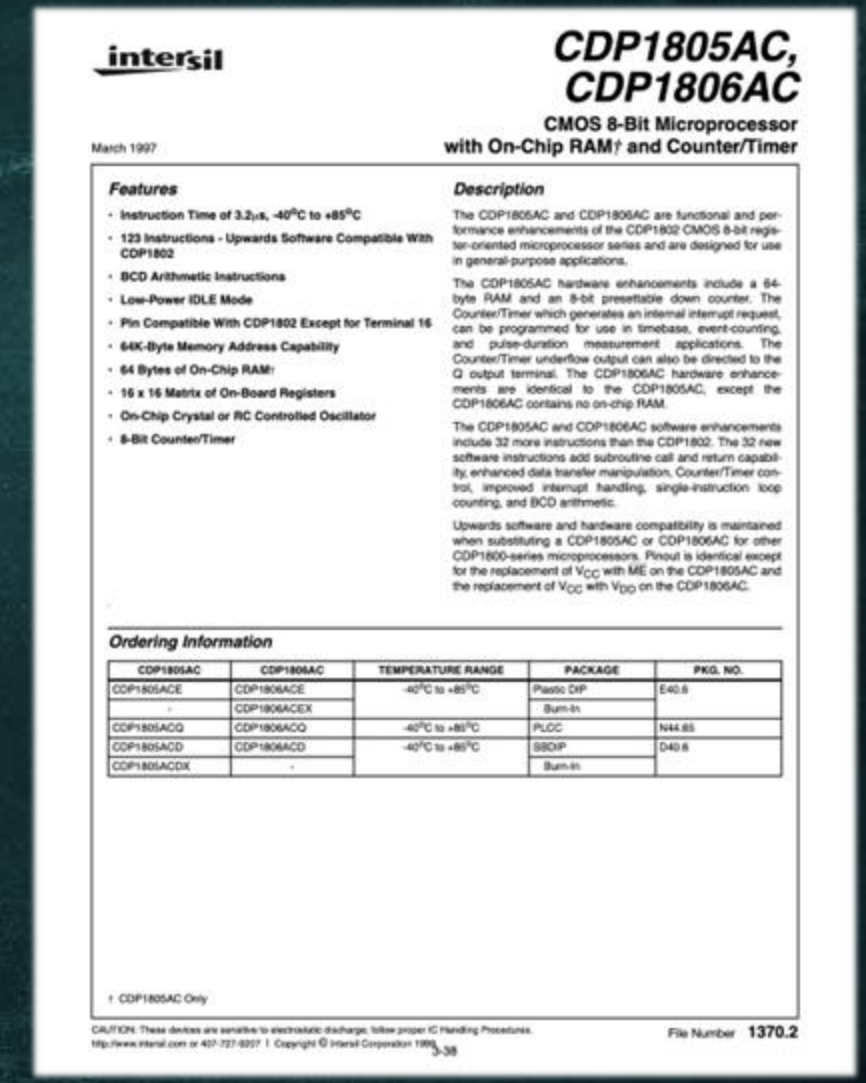
- A total of 12 *additional* phones were rescued.
- There are consequences to dumpster-diving.

TA0042:

Resource Development

An esoteric CPU

- The peculiarities are endless…



intersil

CDP1805AC, CDP1806AC

CMOS 8-Bit Microprocessor with On-Chip RAM† and Counter/Timer

March 1997

Features

- **Instruction Time of 3.2μs, -40°C to +85°C**
- **123 Instructions - Upwards Software Compatible With CDP1802**
- **BCD Arithmetic Instructions**
- **Low-Power IDLE Mode**
- **Pin Compatible With CDP1802 Except for Terminal 16**
- **64K-Byte Memory Address Capability**
- **64 Bytes of On-Chip RAM†**
- **16 x 16 Matrix of On-Board Registers**
- **On-Chip Crystal or RC Controlled Oscillator**
- **8-Bit Counter/Timer**

Description

The CDP1805AC and CDP1806AC are functional and performance enhancements of the CDP1802 CMOS 8-bit register-oriented microprocessor series and are designed for use in general-purpose applications.

The CDP1805AC hardware enhancements include a 64-byte RAM and an 8-bit presettable down counter. The Counter/Timer which generates an internal interrupt request, can be programmed for use in timebase, event-counting, and pulse-duration measurement applications. The Counter/Timer underflow output can also be directed to the Q output terminal. The CDP1806AC hardware enhancements are identical to the CDP1805AC, except the CDP1806AC contains no on-chip RAM.

The CDP1805AC and CDP1806AC software enhancements include 32 more instructions than the CDP1802. The 32 new software instructions add subroutine call and return capability, enhanced data transfer manipulation, Counter/Timer control, improved interrupt handling, single-instruction loop counting, and BCD arithmetic.

Upwards software and hardware compatibility is maintained when substituting a CDP1805AC or CDP1806AC for other CDP1800-series microprocessors. Pinout is identical except for the replacement of V_{CC} with ME on the CDP1805AC and the replacement of V_{CC} with V_{DD} on the CDP1806AC.

Ordering Information

CDP1805AC	CDP1806AC	TEMPERATURE RANGE	PACKAGE	PKG. NO.
CDP1805ACE	CDP1806ACE	-40°C to +85°C	Plastic DIP	E40.6
-	CDP1806ACEX		Burn-In	
CDP1805ACQ	CDP1806ACQ	-40°C to +85°C	PLCC	N44.65
CDP1805ACD	CDP1806ACD	-40°C to +85°C	SBDIP	D40.6
CDP1805ACDX	-		Burn-In	

† CDP1805AC Only

CAUTION: These devices are sensitive to electrostatic discharge; follow proper IC Handling Procedures.
http://www.intersil.com or 407-727-9207 | Copyright © Intersil Corporation 1997

File Number **1370.2**

3-38

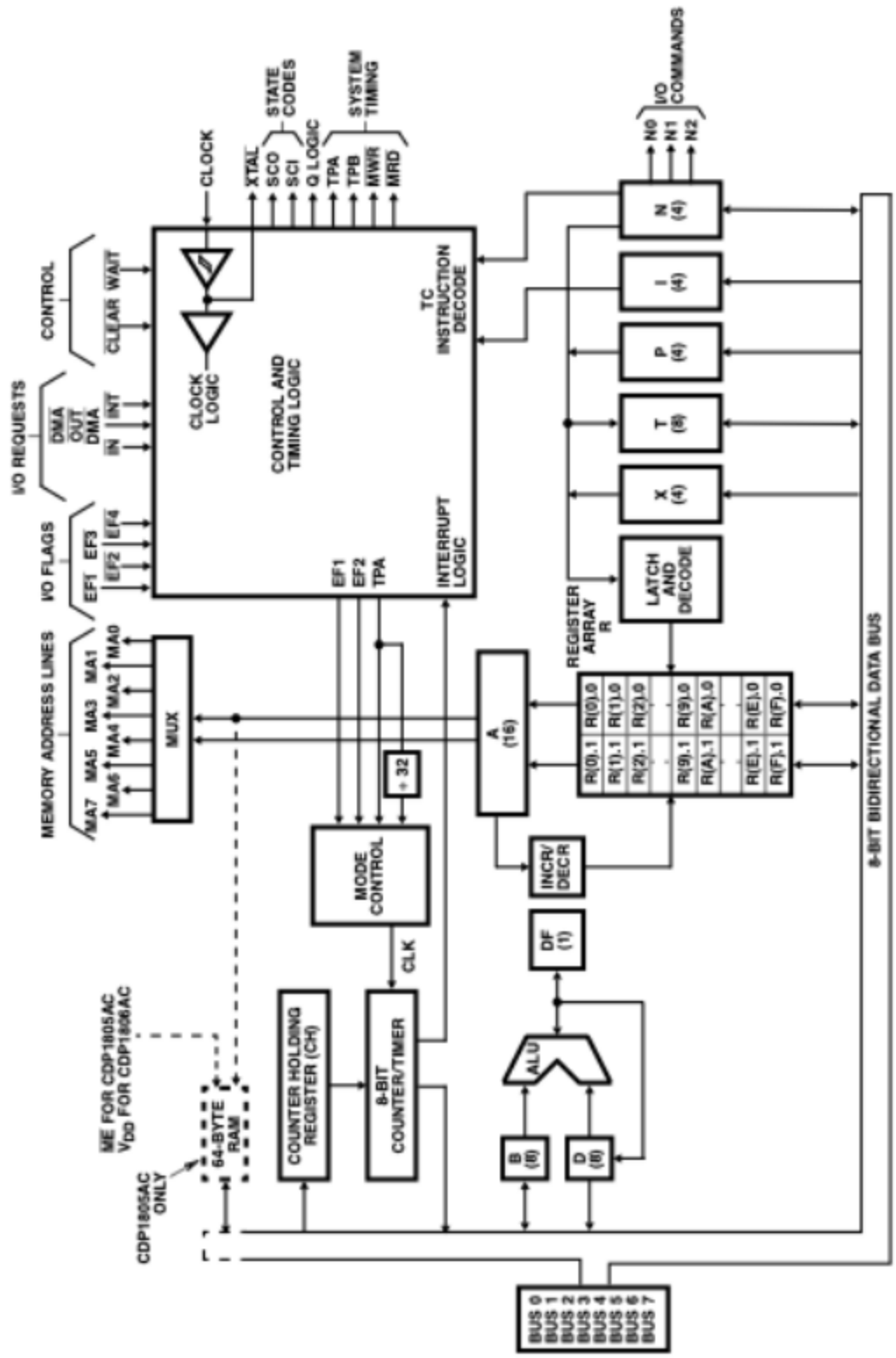


FIGURE 2. BLOCK DIAGRAM FOR CDP1805AC AND CDP1806AC

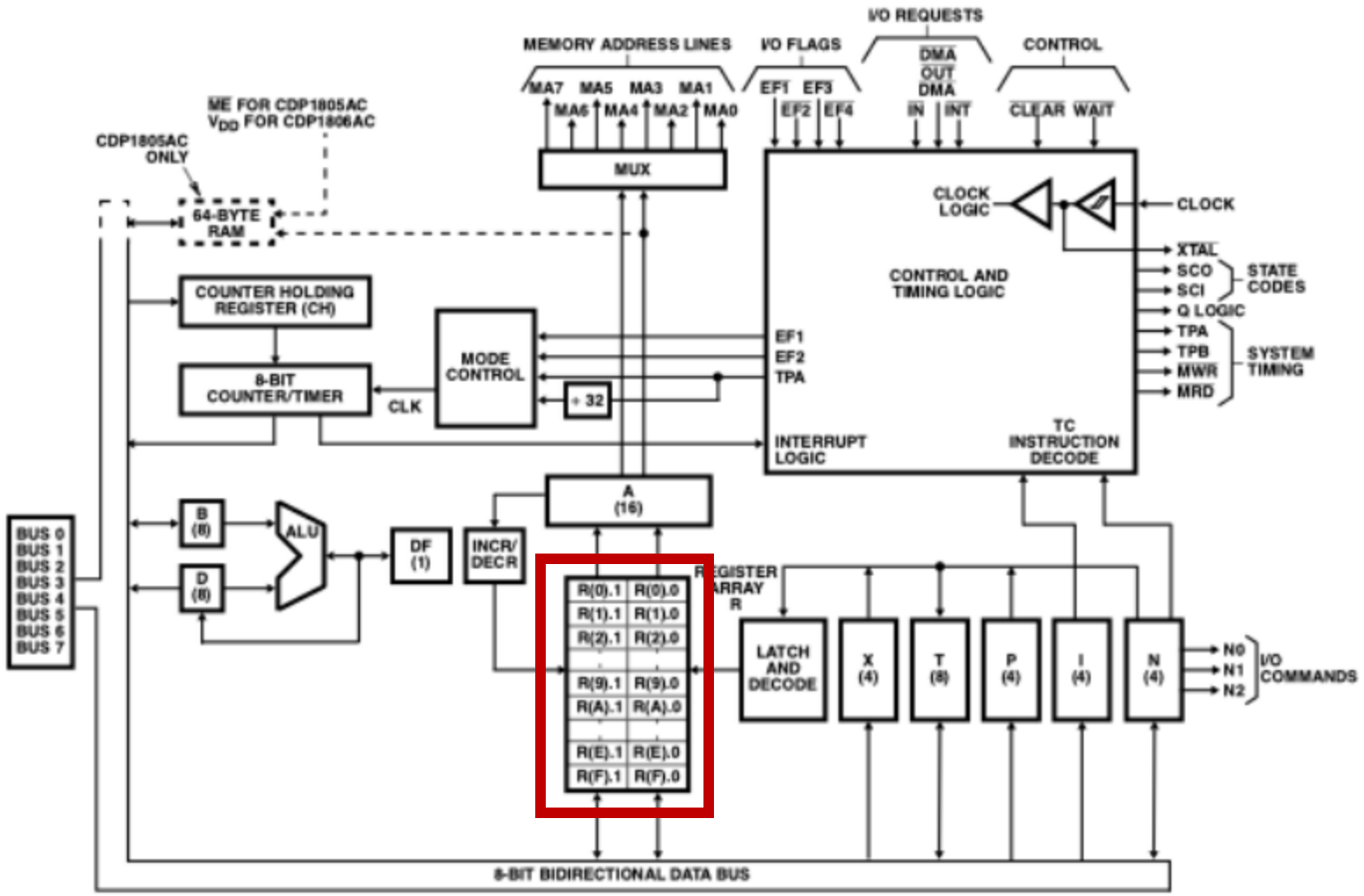


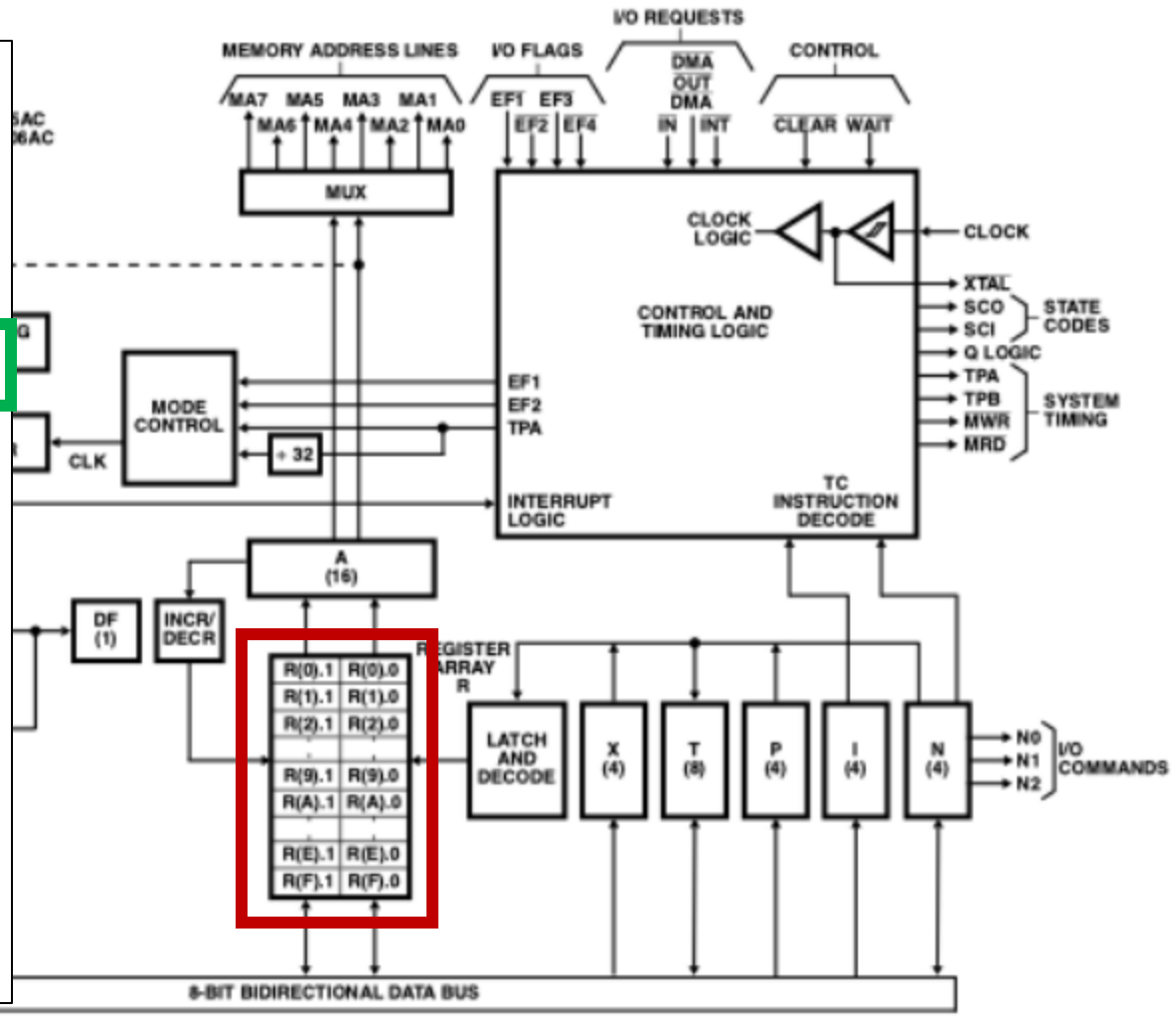
FIGURE 2. BLOCK DIAGRAM FOR CDP1805AC AND CDP1806AC

16 General purpose 16-bit registers (R0..RF)

ANY General purpose register can be the Program Counter

REGISTER SUMMARY

D	8 Bits	Data Register (Accumulator)
DF	1-Bit	Data Flag (ALU Carry)
B	8 Bits	Auxiliary Holding Register
R	16 Bits	1 of 16 Scratch and Registers
P	4 Bits	Designates which Register is Program Counter
X	4 Bits	Designates which Register is Data Pointer
N	4 Bits	Holds Low-Order Instr. Digit
I	4 Bits	Holds High-Order Instr. Digit
T	8 Bits	Holds old X, P after Interrupt (X is high nibble)
Q	1-Bit	Output Flip-Flop
CNTR	8-Bits	Counter/Timer
CH	8 Bits	Holds Counter Jam Value
MIE	1-Bit	Master Interrupt Enable
CIE	1-Bit	Counter Interrupt Enable
XIE	1-Bit	External Interrupt Enable
CIL	1-Bit	Counter Interrupt Latch



CDP1805AC, CDP1806AC

16 General purpose 16-bit registers (R0..RF)

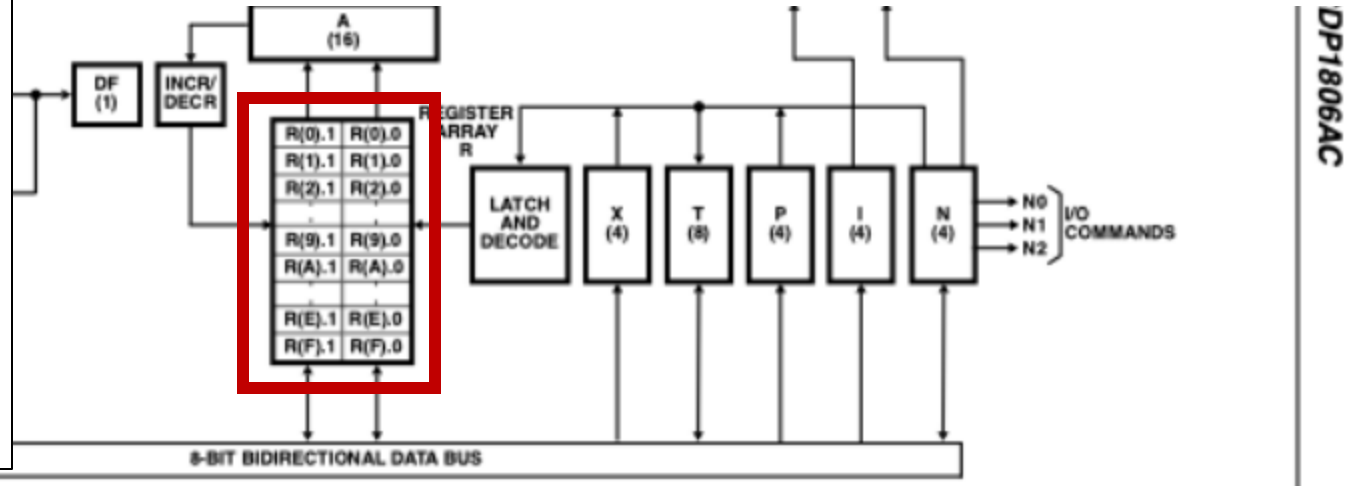
ANY General purpose register can be the Program Counter

REGISTER SUMMARY

D	8 Bits	Data Register (Accumulator)
DF	1-Bit	Data Flag (ALU Carry)
B	8 Bits	Auxiliary Holding Register
R	16 Bits	1 of 16 Scratch and Registers
P	4 Bits	Designates which Register is Program Counter
X	4 Bits	Designates which Register is Data Pointer
N	4 Bits	Holds Low-Order Instr. Digit
I	4 Bits	Holds High-Order Instr. Digit
T	8 Bits	Holds old X, P after Interrupt (X is high nibble)
Q	1-Bit	Output Flip-Flop
CNTR	8-Bits	Counter/Timer
CH	8 Bits	Holds Counter Jam Value
MIE	1-Bit	Master Interrupt Enable
CIE	1-Bit	Counter Interrupt Enable
XIE	1-Bit	External Interrupt Enable
CIL	1-Bit	Counter Interrupt Latch

CALL AND RETURN				
STANDARD CALL	10	SCAL	688N (Note 10)	R(N).0 → M(R(X)); R(N).1 → M(R(X) - 1); R(X) - 2 → R(X); R(P) → R(N); THEN M(R(N)) → R(P).1; M(R(N) + 1) → R(P).0; R(N) + 2 → R(N)
STANDARD RETURN	8	SRET	689N (Note 10)	R(N) → R(P); M(R(X) + 1) → R(N).1; M(R(X) + 2) → R(N).0; R(X) + 2 → R(X)

Immediate arguments to functions are passed *via the bytes after the call*



16 General purpose 16-bit registers (R0..RF)

Searching for a Disassembler

Searching for a Disassembler

IDA Pro supported proces

IIDA Pro supports the processors listed below. ' available in our free SDK.

IDA Pro supported processors

- web assembly (WASM)
- AMD K6-2 3D-Now! extensions
- 32-bit ARM Architecture versions fron instructions and NEON Advanced SIM
  - ARMv4 / ARMv4T: ARM7 cores (A ARM740T), ARM9 cores (ARM9TD
  - ARMv5 / ARMv5TE / ARMv5TEJ: A ARM968E-S/ ARM926EJ-S/ ARM9 ARM1026EJ-S)
  - ARMv6 / ARMv6T2 / ARMv6Z / AR ARM1156T2(F)-S / ARM1176JZ(F)-
  - ARMv6-M: Cortex-M0 / Cortex-M(Freescale Kinetis L and M series, S
  - ARMv7-M: Cortex-M3 (e.g. NXP LI TI Stellaris, Toshiba TX03 / TMPM
  - ARMv7E-M: Cortex-M4 (e.g. NXP LM4F, Freescale Kinetis K series a
  - ARMv7-R: Cortex-R4(F)/Cortex-R5
  - ARMv7-A: Cortex-A5 / Cortex-A7 / Cortex-A15 (e.g. TI Sitara, TI OMAI Nvidia Tegra, Freescale i.MX, Allw
  - ARMv7 (custom): Apple A4/A5/A5 Apple's iPhone/iPod/iPad/AppleTV incomplete; code for any ARM-cor
- ARM64 Architecture (aka AArch64): *
  - ARMv8-A: Cortex-A50/Cortex-A53
  - ARMv8 (custom): Apple A7,A8 etc.
- ARC (Argonaut RISC Core), including code)
- ATMEL AVR (comes with source code)
- DEC PDP-11(comes with source code)
- Fujitsu FR (comes with source code)
- Nintendo Entertainment System (NES) (6502)
- Nintendo Super Entertainment System (SNES) (65
- Nintendo GameBoy (Z80, ARM7)
- Nintendo DS (ARM9, ARM11)
- Hitachi/Renesas H8/300, H8/300L, H8/300H, H8 (comes with source code)H8/330, H8/322, H8/32 H8/336-338, H8/350, H8/3048F, H8/3202, H8/32 H8/3256, H8/3257, H8/3292, H8/3294, H8/3296, H8/3334Y, H8/3336Y, H8/3337Y, H8/3337YF, H8 H8/3534, H8/3434, H8/3434F, H8/3436, H8/343 H8/3613, H8/3614, H8/3712, H8/3713, H8/3714, H H8/3726, H8/3812, H8/3813, H8/3814, H8/3833- H8/3924, H8/3925, H8/3926, H8/3927, H8/3945- H8/3003, H8/3030-32, H8/3048, H8/3070, H8/3 H8S/2245, H8S/2244, H8S/2243, H8S/2242, H8S H8S/2357F, H8S/2143, H8S/2142F, H8S/2345F, H H8S/2235, H8S/2233, H8S/2227, H8S/2225, H8S H8S/2350, H8S/2352, H8S/2340, H8S/2355, H8S H8S/2132FH8S/2655R, H8S/2653R, H8S/2655
- Hitachi H8/500 (comes with source code)
- Hitachi HD 6301, HD 6303, Hitachi HD 64180
- INTEL 8080
- INTEL 8085 (comes with source code)
- INTEL 80196 (comes with source code)
- INTEL 8051 (comes with source code)
- INTEL 860XR (comes with source code)
- INTEL 960 (comes with source code)
- INTEL 80×86 and 80×87 (Intel 8086/8088, 80186 compatibles)
- INTEL Pentium/Celeron/Xeon, including SSE, SSE 512 and other extensions
- Java Virtual Machine (comes with source code)
- KR1878 (comes with source code)
- Microsoft .NET (Common Language Infrastructur
- Mitsubishi MELPS740 or Renesas 740 (comes wit
- Hitachi/Renesas M16C family (M16C/60, M16C/2 M16C/80, M32C/80, R32C/100
- MN102 (comes only with source code)
- MOS Technologies 6502, 65C02 (comes wi
- W65C816S aka 65C816 or 65816 (comes wi
- Motorola/Freescale 68K family: MC680xx,
- Motorola 68xx series: MC6301, MC6303, M MC6808, HCS08, MC6809, MC6811, \
- Motorola MC6812/MC68HC12/CPU12
- Freescale HCS12, HCS12X (including XGATE
- Motorola MC68HC16
- NSC CR16 (comes only with source code)
- NEC V850 series (including V850E1, V850E (comes with source code)
- Renesas RH850 series (RH850G3K, RH850 (comes with source code)
- EFI Byte Code (EBC) (comes with source co
- SPU (Synergistic Processing Unit of the Cel
- MSP430, MSP430X (comes with source co
- Microchip PIC 12XX, PIC 14XX, PIC 18XX, PI
- Microchip 16-bit PIC series (PIC24, dsPIC: F
- Rockwell C39 (comes only with source cod
- Samsung SAM8 (comes with source code)
- SGS Thomson ST-7, and ST-20 (comes with
- TLCS900 (comes only with source code)
- unSP from SunPlus
- Sony SPC700
- Philips XA series (51XA G3)(comes with sou
- RISC-V
- Renesas RL78 series
- Intel xScale
- Z80, Zilog Z8, Zilog Z180, Zilog Z380 (come
- IBM z/Architecture (S390, S390x)
- x64 architecture (aka Intel 64/IA-32e/EM64
- Analog Devices AD218x series (ADSP-2181, 2185(L/M/N), ADSP-2186(L/M/N), ADSP-218 2189M/N)
- Dalvik (Android bytecode, DEX)
- DEC Alpha
- Motorola DSP563xx, DSP566xx, DSP561XX
- TI TMS320C2X, TMS320C5X, TMS320C6X TMS320C55xx, TMS320C3 (comes with so
- TI TMS320C27x/TMS320C28x
- Hewlett-Packard HP-PA aka PA-RISC (come
- Hitachi/Renesas SuperH series. Unexhausti
  - SH-1 core, for example SH7020,SH7021
  - SH-2 core: SH7047, SH7049, SH7105, S
  - SH-2E core: SH7055, SH7058 etc.;
  - SH-2A core: SH7201, SH7206, SH7211, S
  - SH2A-FPU core: SH7450, SH7263 etc.;
  - SH3 core: SH7706, SH7709S;
  - SH4 core: SH7091(Sega Dreamcast), SH
  - SH-4A core: SH7450, SH7451,SH7734, S
- Renesas RXv1, RXv2, RXv3 cores (RX100/RX
- IBM/Motorola PowerPC/POWER architecture
  - Book E (Embedded Controller Instructio
  - Freescale ISA extensions (isel etc.)
  - SPE (Signal Processing Engine) instructi
  - AltiVec (SIMD) instructions
  - Hypervisor and virtualization instruction
  - All instructions from the Power ISA 2.06 Floating Point, Integer Multiply-Accumul
  - Cell BE (Broadband Engine) instructions
  - VLE (Variable Length Encoding) compre
  - Xenon (Xbox 360) instructions, includin
  - Paired Single SIMD instructions (PowerF used in Nintendo Wii and WiiU)
- Motorola/Freescale PowerPC-based cores a limited to):
  - MPC5xx series: MPC533 / MPC535 / MI / MPC563 / MPC564 / MPC566 *Note*: co MPC534/MPC564/MPC556/MPC566 (B not supported
  - MPC8xx series (PowerQUICC): MPC821
  - MPC8xxx series (PowerQUICC II, PowerQUICC II Pro, PowerQUICC III): MPC82xx / MPC83xx / MPC85xx / MPC87xx
  - MPC5xxx series (Qorivva): MPC55xx, MPC56xx, MPC57xx
  - Power PC 4xx, 6xx, 74xx, e200 (including e200z0 with VLE), e500 (including e500v1, e500v2 and e500mc), e600, e700, e5500, e6500 cores
  - QorIQ series: P1, P2, P3, P4, P5 and T1, T2, T4 families
- Infineon Tricore architecture (up to architecture v1.6.2 (AURIX))
- Intel IA-64 Architecture – Itanium.
- Motorola DSP 56K
- MIPS
  - MIPS Mark I (R2000)
  - MIPS Mark II (R3000)
  - MIPS Mark III: (R4000, R4200, R4300, R4400, and R4600)
  - MIPS Mark IV: R8000, R10000, R5900 (Playstation 2)
  - MIPS32, MIPS32r2, MIPS32r3 and MIPS64, MIPS64r2, MIPS64r3
  - Allegrex CPU (Playstation Portable – PSP), including VFPU instructions
  - Cavium Octeon and Octeon2 (cnMIPS) ISA extensions
  - MIPS16 (MIPS16e, MIPS16e2) Application Specific Extension
  - microMIPS Application Specific Extension
  - MIPS-MT, MIPS-3D, smartMIPS Application Specific Extensions
  - Toshiba TX19/TX19A Family Application Specific Extension (MIPS16e+ aka MIPS16e-TX)
- Mitsubishi M32R (comes with source code)
- Mitsubishi M7700 (comes with source code)
- Mitsubishi M7900 (comes with source code)
- Nec 78K0 and Nec 78K0S (comes with source code)
- STMicroelectronics ST9+, ST-10 (comes with source code)
- SPARCII, ULTRASPARC
- Siemens C166 (flow)
- Tensilica Xtensa
- Fujitsu F2MC-16L, Fujitsu F2MC-LC (comes with source code)

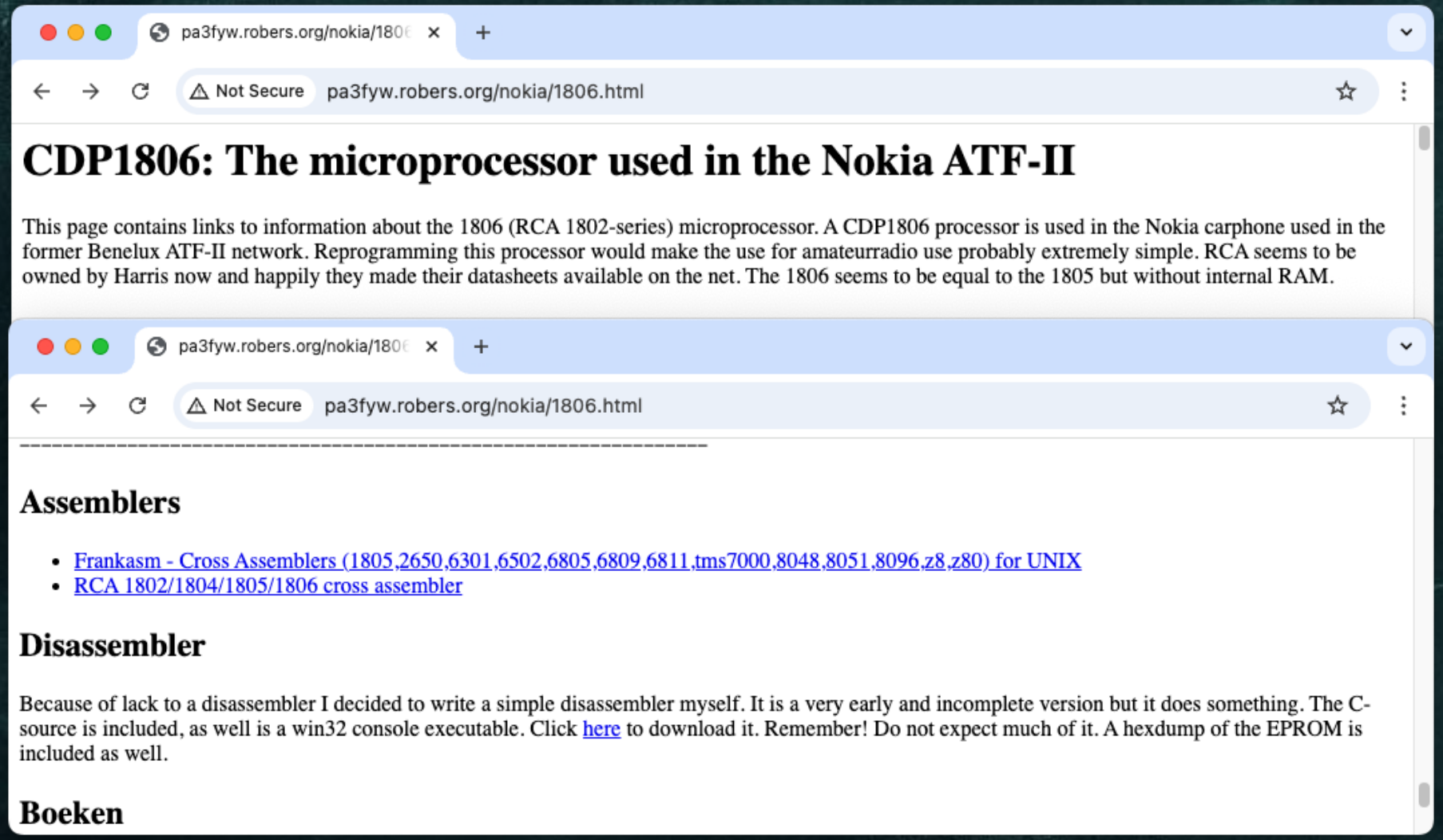
Searching for a Disassembler

Searching for a Disassembler

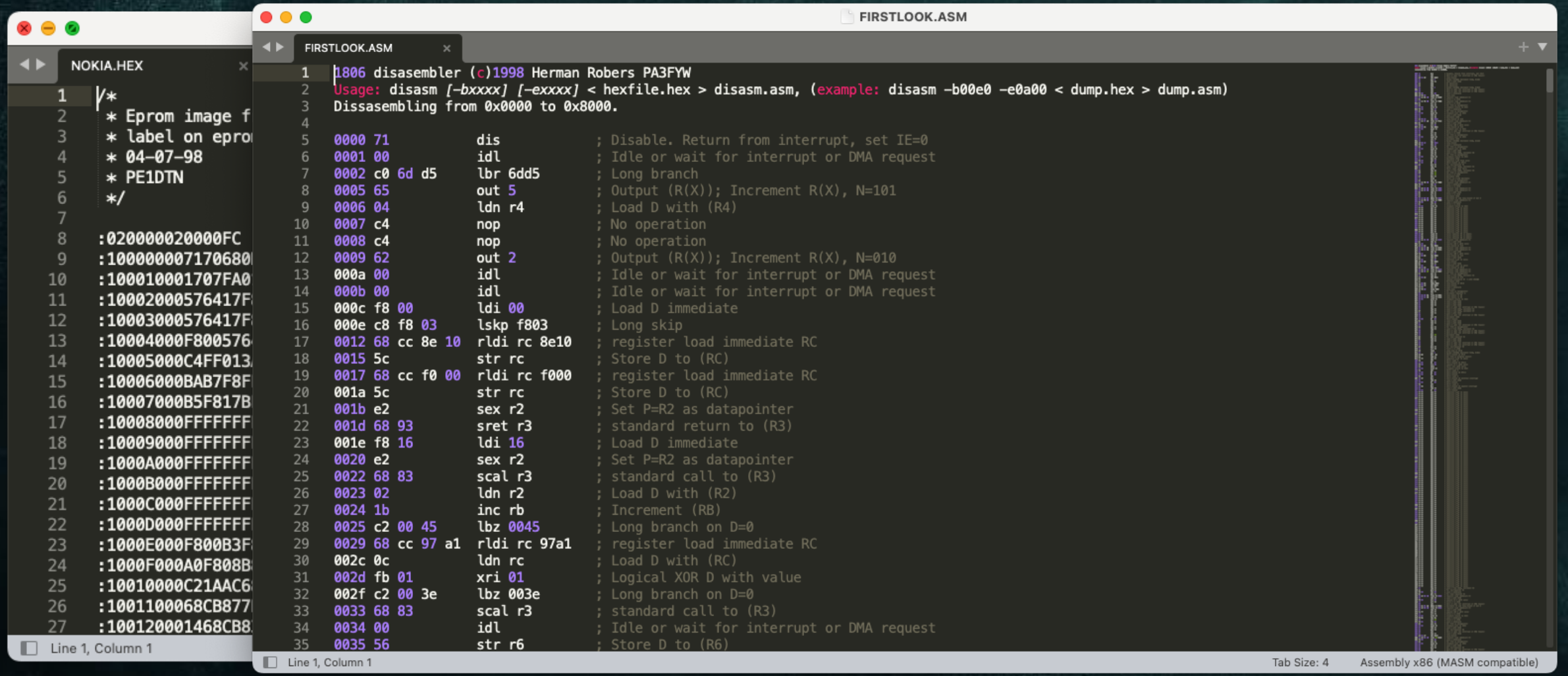
ghidra / Ghidra / **Processors** /

Name	Last commit message
..	
6502	GP-5078: Improvements to Ghidra Module direc
68000	GP-5051: Distinct qemu-system launcher.
8048	GP-5078: Improvements to Ghidra Module direc
8051	GP-5078: Improvements to Ghidra Module direc
8085	GP-5078: Improvements to Ghidra Module direc
AARCH64	GP-2432 golang api snapshot, generics, closur
ARM	Merge remote-tracking branch 'origin/patch'
Atmel	GP-5196 Adding support for expressions and sy
BPF	GP-5078: Improvements to Ghidra Module direc
CP1600	GP-5078: Improvements to Ghidra Module direc
CR16	GP-5078: Improvements to Ghidra Module direc
DATA	GP-5078: Improvements to Ghidra Module direc
Dalvik	GP-0: Certify
HCS08	GP-5078: Improvements to Ghidra Module direc
HCS12	GP-5078: Improvements to Ghidra Module direc
JVM	GP-0: Cleaning up certain types of javadoc erro
Loongarch	GP-5051: Distinct qemu-system launcher.
M16C	GP-5078: Improvements to Ghidra Module directory layout	6 months ago
M8C	GP-5078: Improvements to Ghidra Module directory layout	6 months ago
MC6800	GP-5078: Improvements to Ghidra Module directory layout	6 months ago
MCS96	GP-5078: Improvements to Ghidra Module directory layout	6 months ago
MIPS	GP-5406 Correct ELF import issue for MIPS debug file which fails on	2 months ago
PA-RISC	GP-5078: Improvements to Ghidra Module directory layout	6 months ago
PIC	Merge branch 'GP-0_ryanmkurtz_PR-6204_antoniovazquezblanco_pic'	last month
PowerPC	GP-5411 Added plt thunk patterns, pre-fill of edit thunk GUI, loosened	last month
RISCV	GP-5051: Distinct qemu-system launcher.	4 months ago
Sparc	GP-5051: Distinct qemu-system launcher.	4 months ago
SuperH	GP-5078: Improvements to Ghidra Module directory layout	6 months ago
SuperH4	GP-5051: Distinct qemu-system launcher.	4 months ago
TI_MSP430	GP-5189 Add range attributes to VarargsFilter	4 months ago
Toy	GP-4643: Add a JIT-accelerated p-code emulator (API/scripting only)	3 months ago
V850	GP-5078: Improvements to Ghidra Module directory layout	6 months ago
Xtensa	GP-5051: Distinct qemu-system launcher.	4 months ago
Z80	GP-5078: Improvements to Ghidra Module directory layout	6 months ago
eBPF	GP-5078: Improvements to Ghidra Module directory layout	6 months ago
tricore	Merge remote-tracking branch 'origin/patch'	last month
x86	Merge remote-tracking branch 'origin/patch'	3 weeks ago

Searching for a Disassembler



Searching for a Disassembler



Writing a CPU Module

IDA-Pro

- C++ SDK Development
- IDAPython alternative
- I don’t own a recent version

Ghidra

- Open-source project
- Text-file definition
- It’s free

Writing a CPU Module

- RTFM.

intersil

CDP1805AC, CDP1806AC

CMOS 8-Bit Microprocessor with On-Chip RAM† and Counter/Timer

March 1997

Features

- Instruction Time of 3.2µs, -40°C to +85°C
- 123 Instructions - Upwards Software Compatible With CDP1802
- BCD Arithmetic Instructions
- Low-Power IDLE Mode
- Pin Compatible With CDP1802 Except for Terminal 16
- 64K-Byte Memory Address Capability
- 64 Bytes of On-Chip RAM†
- 16 x 16 Matrix of On-Board Registers
- On-Chip Crystal or RC Controlled Oscillator
- 8-Bit Counter/Timer

Description

The CDP1805AC and CDP1806AC are functional and performance enhancements of the CDP1802 CMOS 8-bit register-oriented microprocessor series and are designed for use in general-purpose applications.

The CDP1805AC hardware enhancements include a 64-byte RAM and an 8-bit presettable down counter. The Counter/Timer which generates an internal interrupt request, can be programmed for use in timebase, event-counting, and pulse-duration measurement applications. The Counter/Timer underflow output can also be directed to the Q output terminal. The CDP1806AC hardware enhancements are identical to the CDP1805AC, except the CDP1806AC contains no on-chip RAM.

The CDP1805AC and CDP1806AC software enhancements include 32 more instructions than the CDP1802. The 32 new software instructions add subroutine call and return capability, enhanced data transfer manipulation, Counter/Timer control, improved interrupt handling, single-instruction loop counting, and BCD arithmetic.

Upwards software and hardware compatibility is maintained when substituting a CDP1805AC or CDP1806AC for other CDP1800-series microprocessors. Pinout is identical except for the replacement of V_{CC} with ME on the CDP1805AC and the replacement of V_{CC} with V_{DD} on the CDP1806AC.

Ordering Information

CDP1805AC	CDP1806AC	TEMPERATURE RANGE	PACKAGE	PKG. NO.
CDP1805ACE	CDP1806ACE	-40°C to +85°C	Plastic DIP	E40.6
-	CDP1806ACEX		Burn-In	
CDP1805ACQ	CDP1806ACQ	-40°C to +85°C	PLCC	N44.65
CDP1805ACD	CDP1806ACD	-40°C to +85°C	SBDIP	D40.6
CDP1805ACDX	-		Burn-In	

† CDP1805AC Only

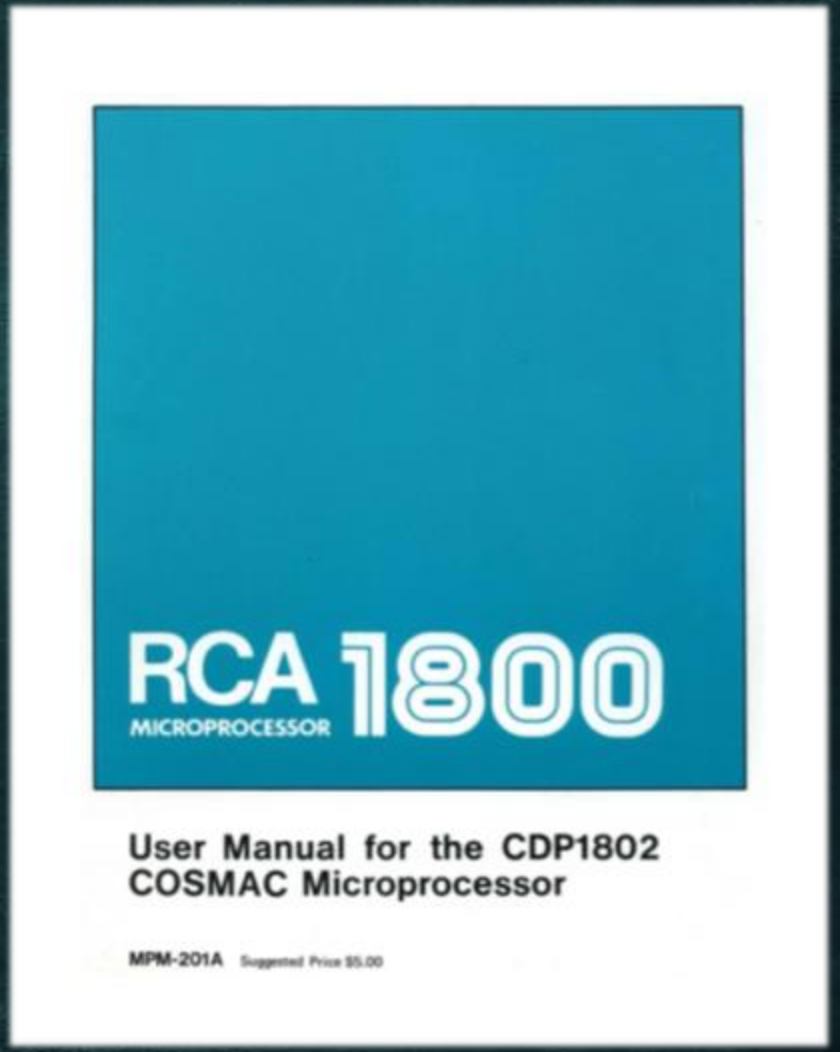
CAUTION: These devices are sensitive to electrostatic discharge; follow proper IC Handling Procedures.
http://www.intersil.com or 407-727-9207 | Copyright © Intersil Corporation 1999

File Number 1370.2

3-38

Writing a CPU Module

- RTFM.
- RTFM some more.



Writing a CPU Module

- RTFM.
- RTFM some more.
- Read user-generated content.

COSMAC ELF

The CDP1802's Place in Microcomputing History

A Short Course In Programming by Tom Pittman

Chapter 0 -- How To Read This Book

Programming a computer is like driving a car -- you learn how to do it by doing it, not by reading a book. But, also like learning to drive, it helps to have someone show you which knobs to twist and how to point the car (or program) in the direction you wish to go. That is the function of this book. I will try to show you how your computer works, so you can learn to write programs yourself.

Driving a car takes muscles (but not much) to turn the steering wheel, push the pedals, and so on. Programming a computer takes brains (but not much) to select the right instructions, anticipate the computer's response, and so on. Very few people do not have the necessary muscles to drive a car. Very few people do not have the necessary brains to program a computer. You probably can do both (at least after you finish learning what is in this book and if you already know how to drive).

You do not learn how to drive a car by sitting in your easy chair and waving your pinky in the air. You cannot learn how to program a computer by sitting in your easy chair and "speed-reading" this book. Plan on taking a couple hours or even a couple evenings on each chapter (except the first two, which are not very difficult).

Writing a CPU Module

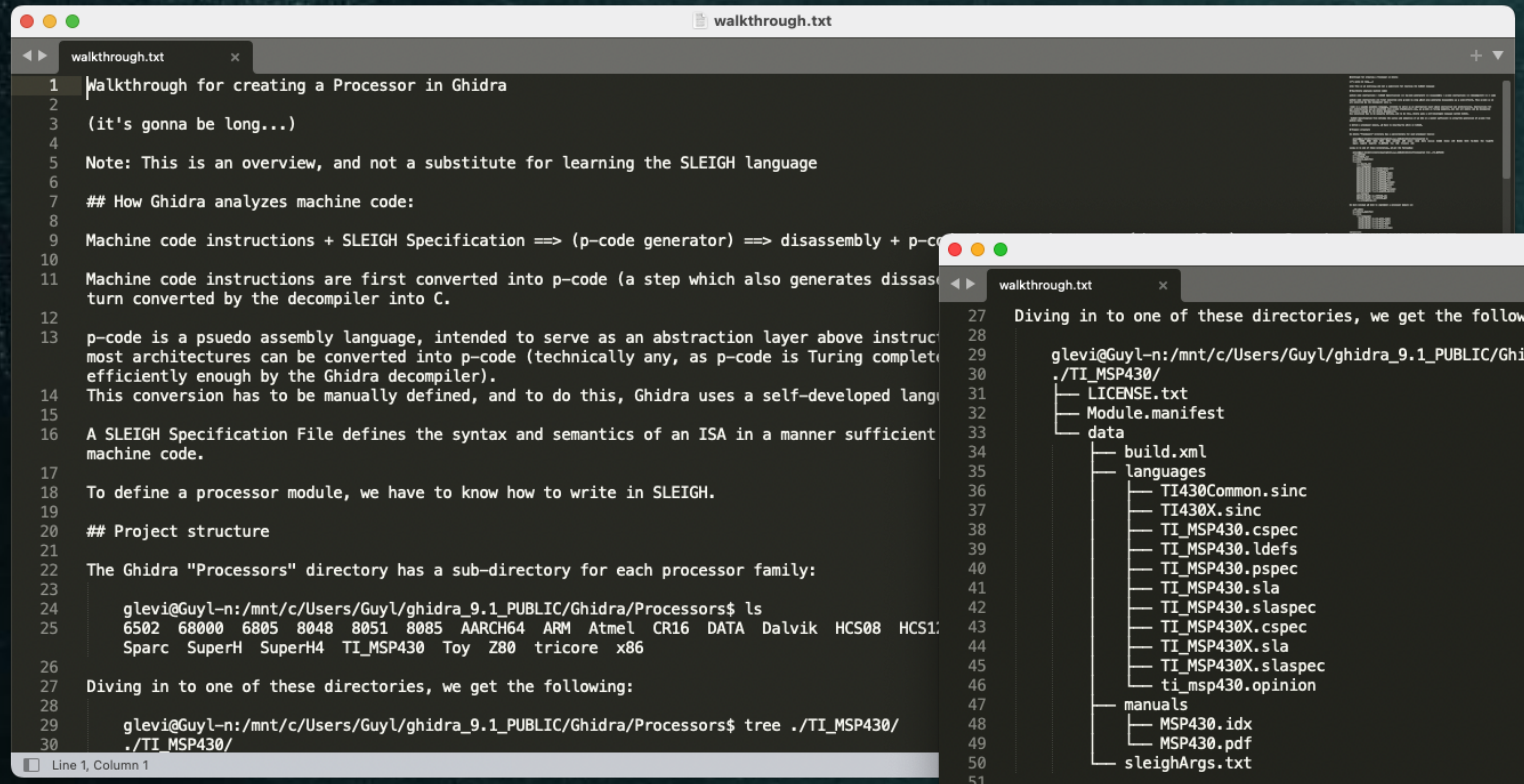
- Infiltrate 2019 preso on Ghidra:

SLEIGH

- Ghidra's language for describing instruction sets to facilitate RE
- **Disassembly:** translate bit-encoded machine instructions into human-readable assembly language statements
- **Semantics:** translate machine instructions into p-code instructions (one-to-many) for decompilation, analysis, and emulation
- Based off of SLED (Specification Language for Encoding and Decoding), a 1997 academic IL

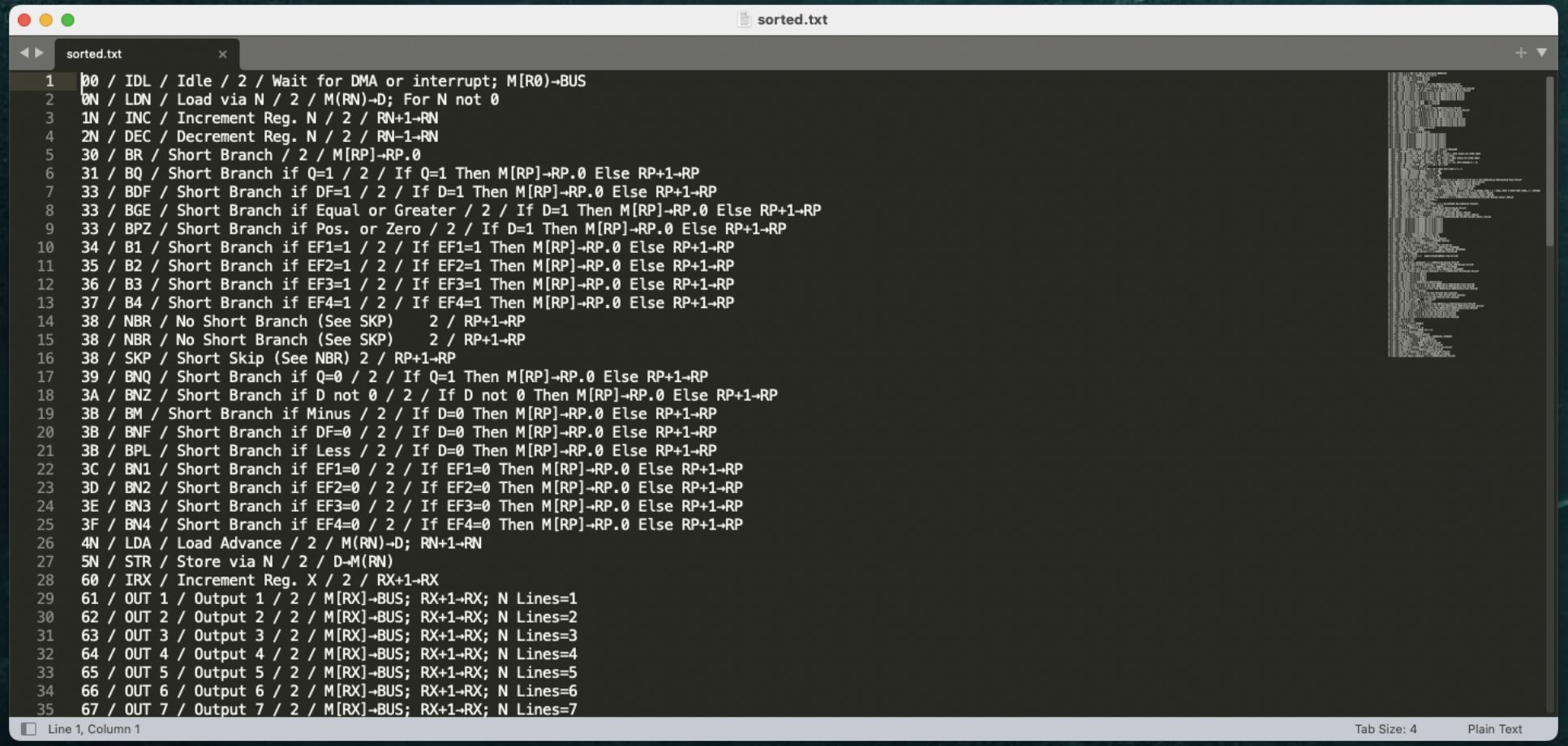
Writing a CPU Module

- Used the MSP430 definition as an example to follow.



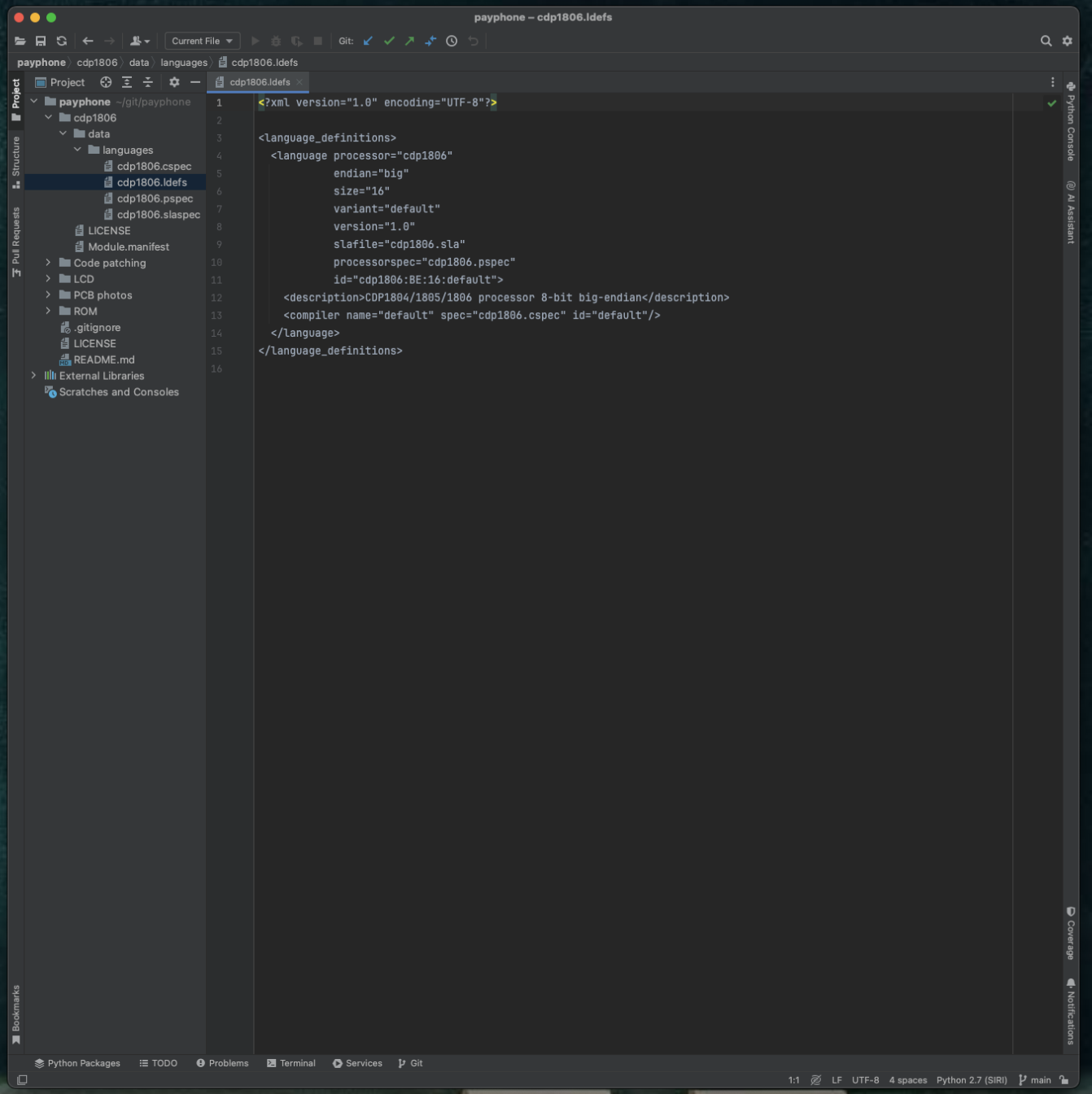
Writing a CPU Module

- Prepared a reference for all instructions.



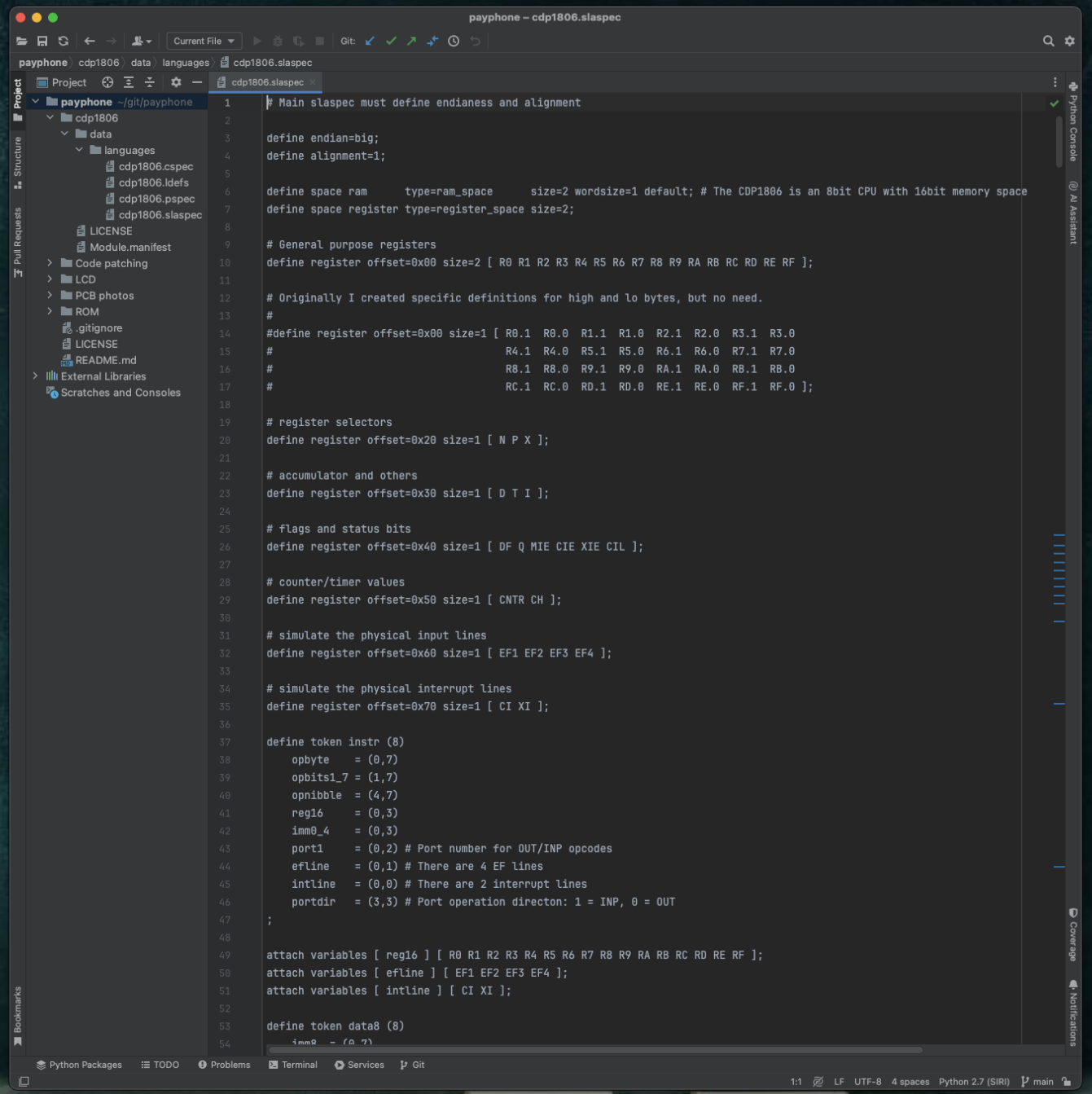
Writing a CPU Module

- Language definitions.



Writing a CPU Module

- Language definitions.
- Architecture definitions.



Writing a CPU Module

- Language definitions.
- Architecture definitions.
- Pseudo-instructions and macros.