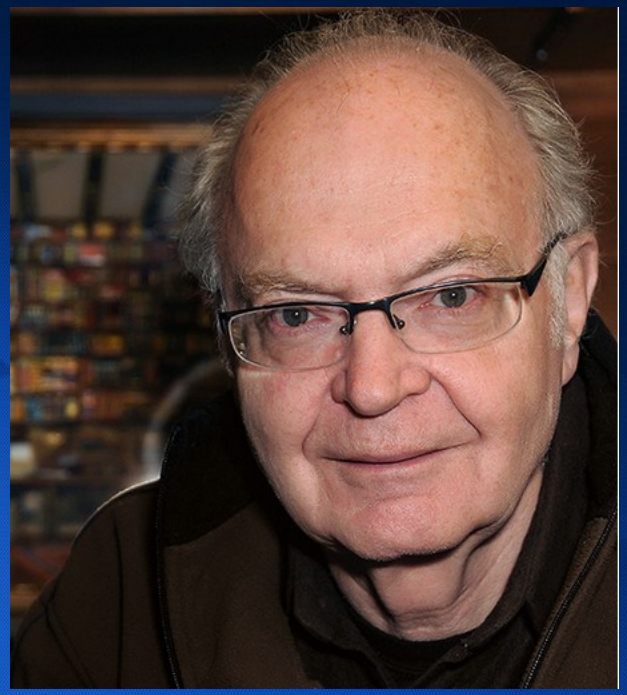
CSU33D01 Microprocessor Systems I

Assessment structure

Semester 1 exam 100% In person written 2 hour examination.

Supplemental exam 100% In person written 2 hour examination.

https://www.scss.tcd.ie/John.Waldron/CSU33D01/CSU33D01.html

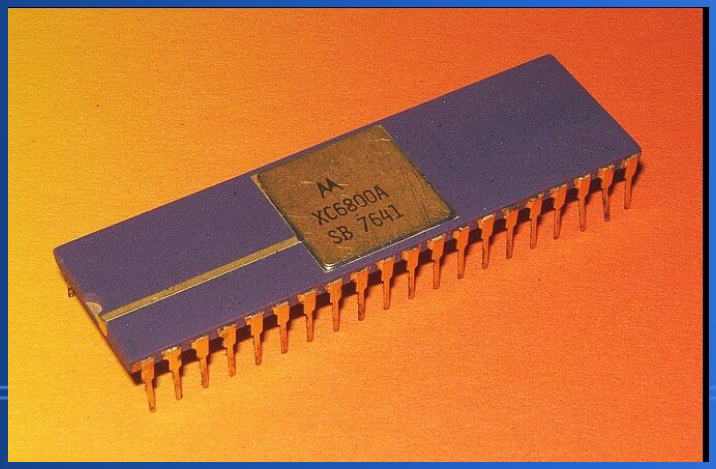


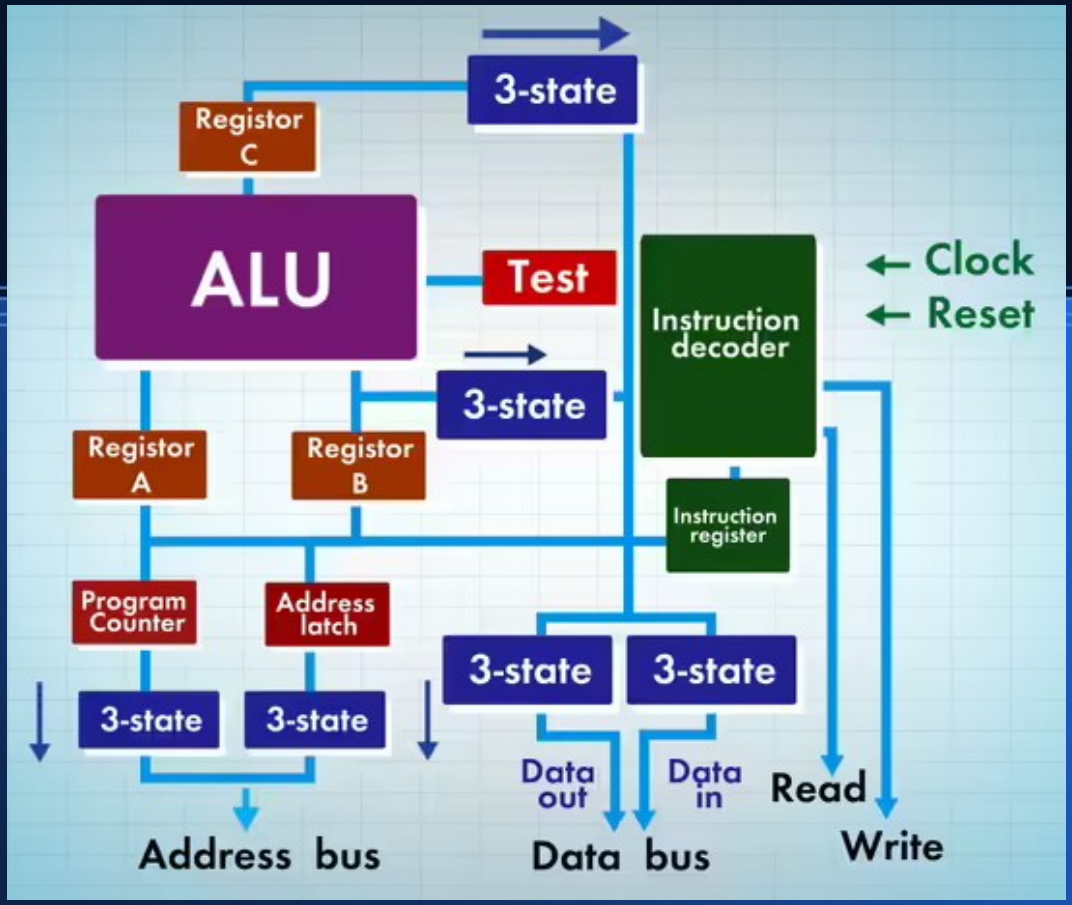
Donald Ervin Knuth is the author of
the multi-volume work
The Art of Computer Programming

"A person who is more than casually interested in computers
should be well schooled in machine language, since it is a
fundamental part of a computer"

https://en.wikipedia.org/wiki/Donald_Knuth

A microprocessor is a computer processor for which the data processing logic and control is included on a single integrated circuit (IC), or a small number of ICs. The microprocessor contains the arithmetic, logic, and control circuitry required to perform the functions of a computer's central processing unit (CPU). The microprocessor is capable of interpreting and executing machine code and performing arithmetic operations. The microprocessor is a multipurpose, clock-driven, register-based, digital integrated circuit that accepts binary data as input, processes it according to instructions stored in its memory, and provides results (also in binary form) as output.



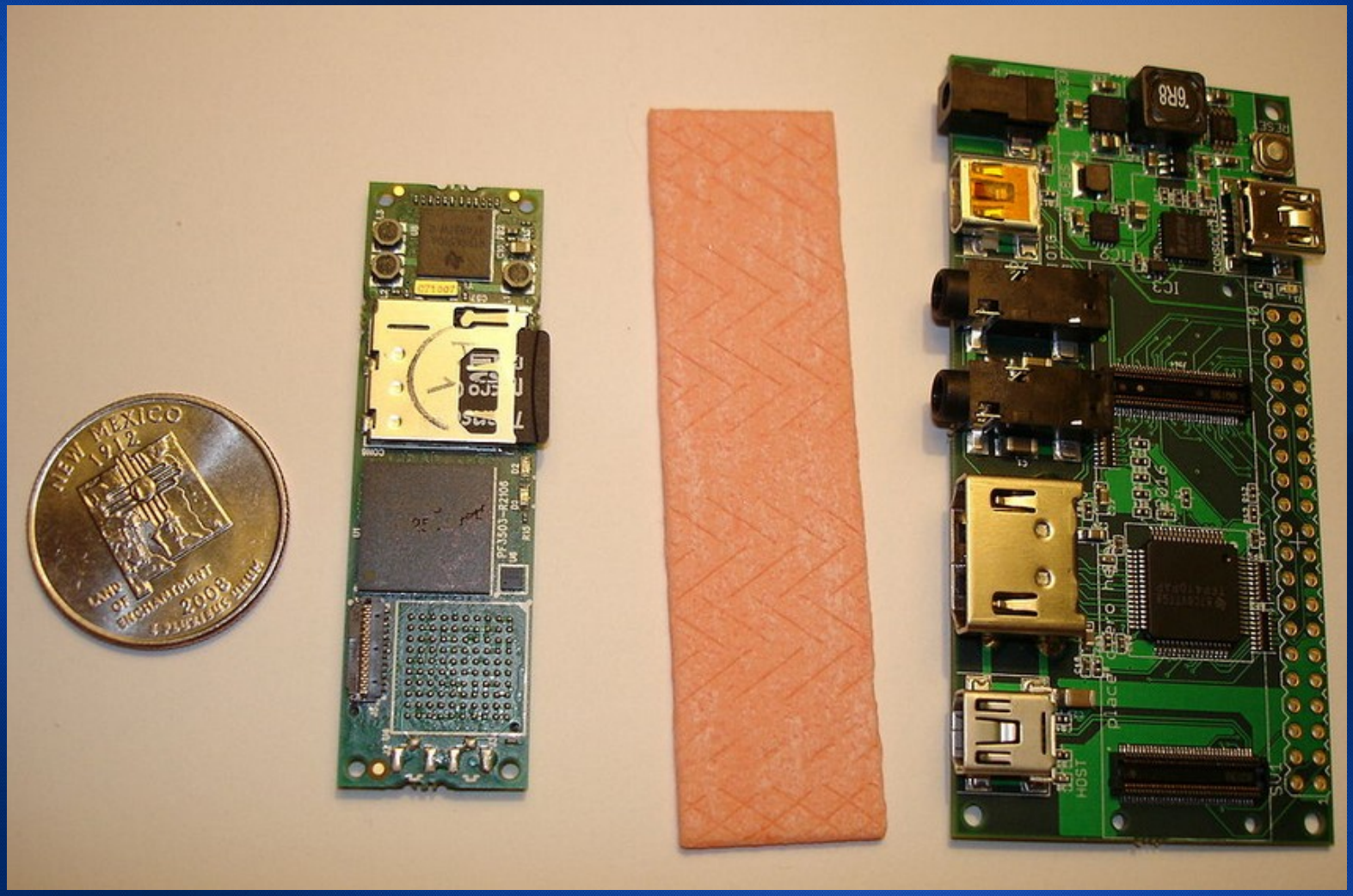


The test register is a special latch that can hold values from comparisons

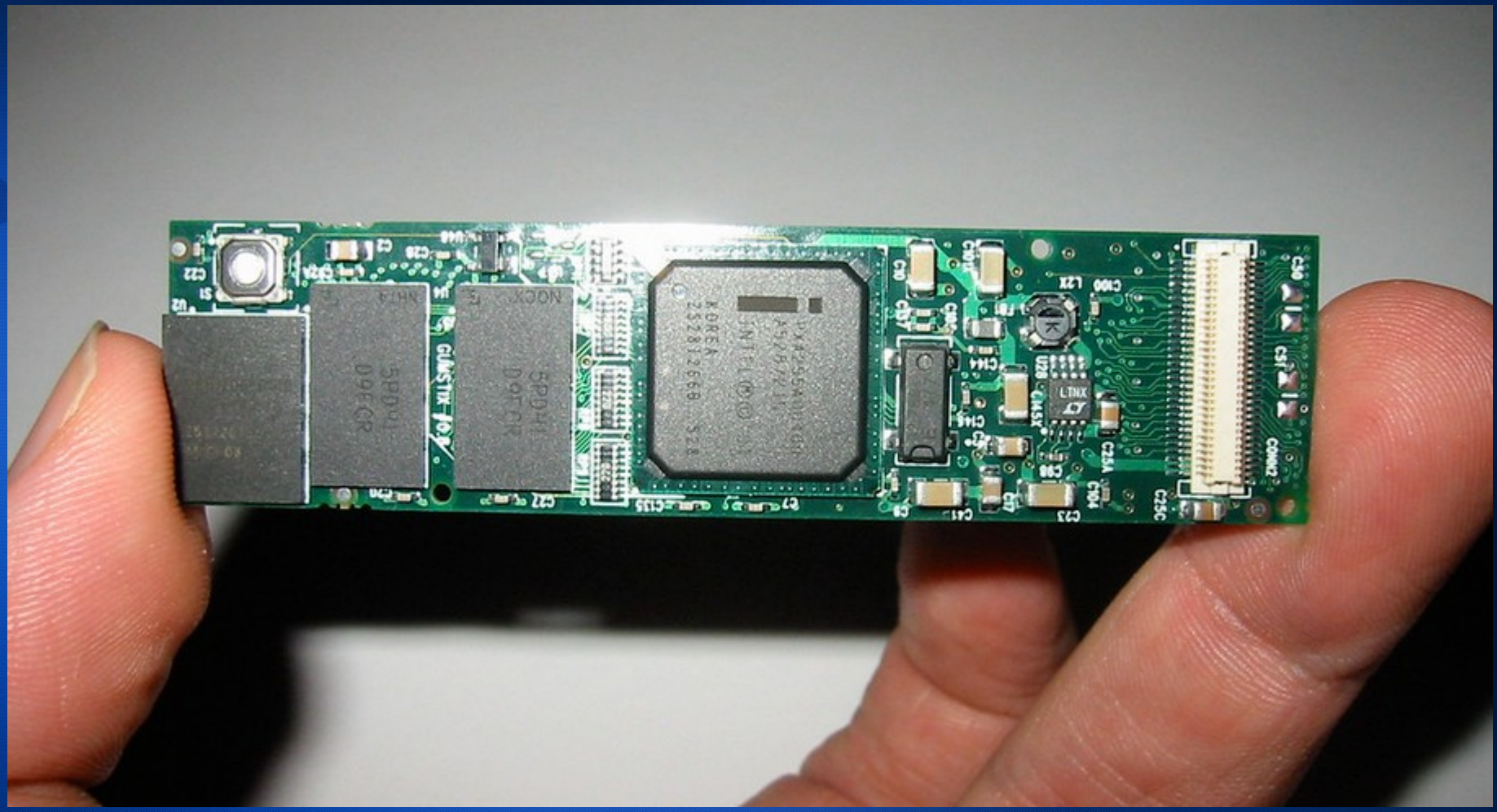
Using its ALU (Arithmetic/Logic Unit), a microprocessor can perform mathematical operations like addition, subtraction, multiplication and division.
Modern microprocessors contain complete floating-point processors FPU that can perform sophisticated operations on large floating-point numbers.
A microprocessor can move data from one memory location to another.
A microprocessor can make decisions and jump to a new set of instructions based on those decisions.

Gumstix is an American multinational corporation headquartered in Redwood City, California. It develops and manufactures small system boards comparable in size to a stick of gum.

A side-by-side size comparison of a US Quarter, a Gumstix Overo Earth, a stick of gum, and the Gumstix Summit expansion board.



Gumstix connex 400MHz



A system on a chip or system-on-chip (SoC) is an integrated circuit that integrates most or all components of a computer or other electronic system. An SoC integrates a microcontroller, microprocessor or perhaps several processor cores with peripherals like a GPU, Wi-Fi and cellular network radio modems, and/or one or more coprocessors.

https://en.wikipedia.org/wiki/System_on_a_chip

PXA System on a Chip (SoC) products were designed in Austin, Texas. The code-names for this product line are small towns in Texas, primarily near deer hunting leases frequented by the Intel XScale core and mobile phone SoC marketing team.

	Released date	Code name	ARM type	Fab	Size	Pin coun	Clock speeds
PXA21x and PXA26x							
PXA210	February 2002	-	ARMv5 XScale1 core[4]		13x13mm[5]	255 pin T-PBGA	133 MHz and 200 MHz
PXA250		**Cotulla**			17x17mm[5]	256-pin PBGA	200 MHz, 300 MHz and 400 MHz[6]
PXA255	March 2003						

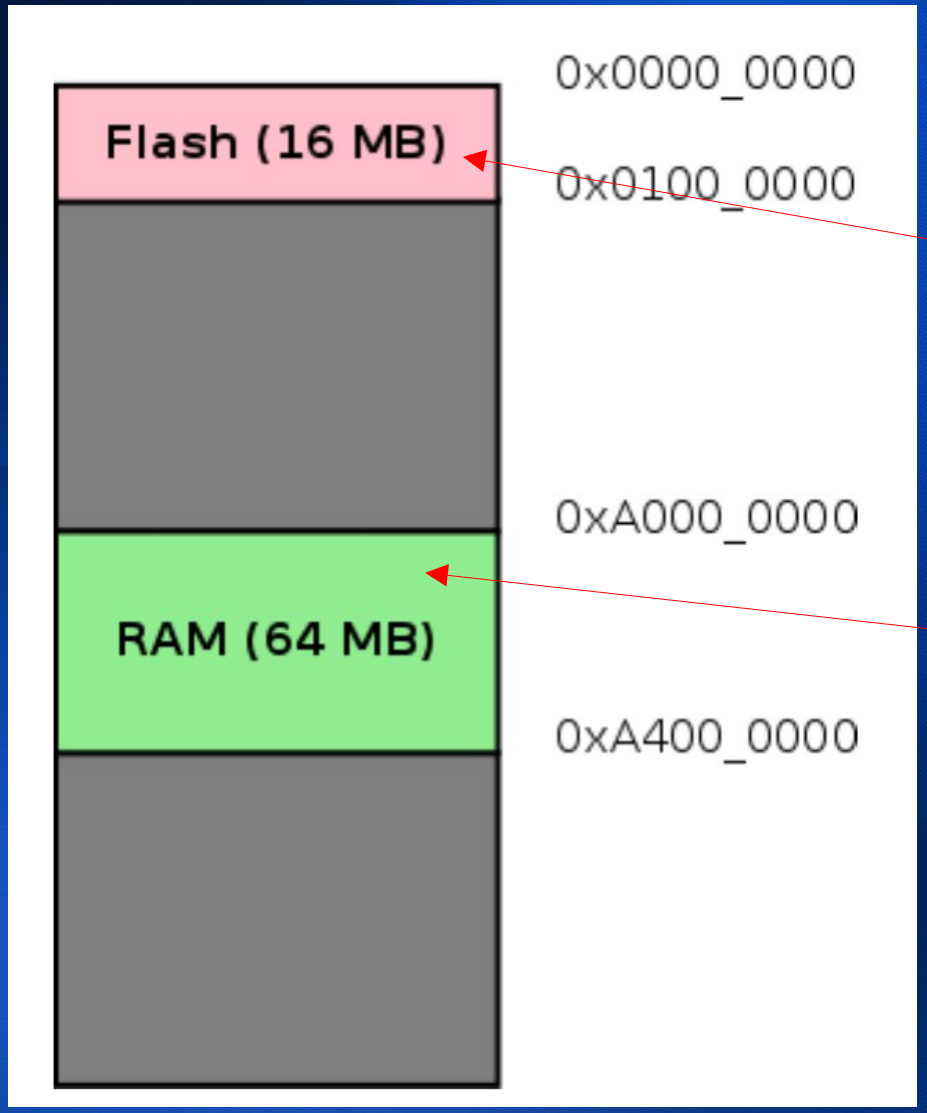
XScale microprocessors can be found in products such as the popular RIM BlackBerry handheld, the Dell Axim family of Pocket PCs, most of the Zire, Treo and Tungsten Handheld lines by Palm, later versions of the Sharp Zaurus, the Motorola A780, the Acer n50, the Compaq iPaq 3900 series and many other PDAs. It is used as the main CPU in the Iyonix PC desktop computer running RISC OS, and the NSLU2 (Slug) running a form of Linux. The XScale is also used in devices such as PVPs (Portable Video Players), PMCs (Portable Media Centres), including the Creative Zen Portable Media Player and Amazon Kindle E-Book reader, and industrial embedded systems. At the other end of the market, the XScale IOP33x Storage I/O processors are used in some Intel Xeon-based server platforms.

This is the ARM version we use throughout this module

Architecture	Core bit-width	Cores	
		Arm Ltd.	Third-party
ARMv1	32	ARM1	
ARMv2	32	ARM2, ARM250, ARM3	Amber, STORM Open Soft Core[60]
ARMv3	32	ARM6, ARM7	
ARMv4	32	ARM8	StrongARM, FA526, ZAP Open Source Processor Core
ARMv4T	32	ARM7TDMI, ARM9TDMI, SecurCore SC100	
ARMv5TE	32	ARM7EJ, ARM9E, ARM10E	XScale, FA626TE, Feroceon, PJ1/Mohawk
ARMv6	32	ARM11	
ARMv6-M	32	ARM Cortex-M0, ARM Cortex-M0+, ARM Cortex-M1, SecurCore SC000	
ARMv7-M	32	ARM Cortex-M3, SecurCore SC300	Apple M7
ARMv7E-M	32	ARM Cortex-M4, ARM Cortex-M7	
ARMv8-M	32	ARM Cortex-M23,[62] ARM Cortex-M33[63]	
ARMv7-R	32	ARM Cortex-R4, ARM Cortex-R5, ARM Cortex-R7, ARM Cortex-R8	
ARMv8-R	32	ARM Cortex-R52	
	64	ARM Cortex-R82	
ARMv7-A	32	ARM Cortex-A5, ARM Cortex-A7, ARM Cortex-A8, ARM Cortex-A9, ARM Cortex-A12, ARM Cortex-A15, ARM Cortex-A17	Qualcomm Scorpion/Krait, PJ4/Sheeva, Apple Swift (A6, A6X)
ARMv8-A	32	ARM Cortex-A32[68]	
	64/32	ARM Cortex-A35,[69] ARM Cortex-A53, ARM Cortex-A57,[70] ARM Cortex-A72,[71] ARM Cortex-A73[72]	X-Gene, Nvidia Denver 1/2, Cavium ThunderX, AMD K12, Apple Cyclone (A7)/Typhoon (A8, A8X)/Twister (A9, A9X)/Hurricane+Zephyr (A10, A10X), Qualcomm Kryo, Samsung M1/M2 ("Mongoose") /M3 ("Meerkat")
	64	ARM Cortex-A34[79]	

Gumstix Connex (PXA255) Memory Map

XScale uses a Harvard memory architecture with separate buses for instructions and data.



CPUlator Computer System Simulator

CPUlator is a Nios II, ARMv7, MIPS, and RISC-V RV32 simulator of a computer system (processor and I/O devices) and debugger that runs in a modern web browser.

It is designed as a tool for learning assembly-language programming and computer organization.

ARMv7 Generic

This is a ARMv7 CPU with 4 GB of memory and no other I/O devices.

WebArm Submit Code

Assemble

```
        MOV     r1, #0x8
        MOV     r2, #0x6
        LDR     r11, =0xA1000400        @ ptr = String
        STRB    r1, [r11], #1           @ Memory.Byte[ptr++] = ch
        STRB    r2, [r11], #1           @ Memory.Byte[ptr++] = ch


stop:   B       stop

str:    .asciz "2G3niwo"

```

WebArm uses GNU ARM Assembly, ARMv5TE Architecture and runs programs on a simulated Gumstix Connex (PXA255). Code written in other assembly languages (e.g. Keil) or for other ARM architectures or for other boards will not work.

new step continue restart R0 set reg breakpoint

```
R00=00000000 R01=00000000 R02=00000000 R03=00000000  > 3 0000 0810A0E3                MOV     r1, #0x8
R04=00000000 R05=00000000 R06=00000000 R07=00000000    4 0004 0620A0E3                MOV     r2, #0x6
R08=00000000 R09=00000000 R10=00000000 R11=00000000    5 0008 10B09FE5                LDR     r11, =0xA1000400
R12=00000000 R13=00000000 R14=00000000 R15=00000000    6 000c 0110CBE4                STRB    r1, [r11], #1
PSR=400001d3 -Z--                                      7 0010 0120CBE4                STRB    r2, [r11], #1
                                                      11 0014 FEFFFFEA        stop:   B       stop
0xA1000400: 00 00 00 00 00 00 00 00                   13 0018 3247336E        str:    .asciz "2G3niwo"
0xA1000408: 00 00 00 00 00 00 00 00                   13      69776F00
0xA1000410: 00 00 00 00 00 00 00 00                   17 0020 000400A1
0xA1000418: 00 00 00 00 00 00 00 00
0xA1000420: 00 00 00 00 00 00 00 00
0xA1000428: 00 00 00 00 00 00 00 00
0xA1000430: 00 00 00 00 00 00 00 00
0xA1000438: 00 00 00 00 00 00 00 00
```

Recommended textbook (covers the popular ARM7TDMI, but also addresses the latest architectures from ARM, including Cortex™-A, Cortex-R, and Cortex-M processors—all of which have slightly different instruction sets, programmer's models, and exception handling.)

ARM Assembly Language
Fundamentals and Techniques, Second Edition
By William Hohl, Christopher Hinds