

# Computer Architecture LU

## Processor Comparison

Alexandra Schuster	<code>alexandra.schuster@student.tuwien.ac.at</code>
Franz Hartl	<code>FHartl.stud@gmx.net</code>
Harald Weillechner	<code>haraldweillechner@gmx.at</code>
Peter Hilber	<code>peter.hilber@student.tuwien.ac.at</code>

October 16, 2007

# Contents

<b>1</b>	<b>MOS Technology 6502</b>	<b>3</b>
1.1	History . . . . .	3
1.2	Registers . . . . .	3
1.3	Instruction Set . . . . .	4
1.3.1	Addressing Modes . . . . .	4
1.3.2	Arithmetic Instructions . . . . .	4
1.3.3	Compare Instructions . . . . .	4
1.3.4	Register Instructions . . . . .	4
1.3.5	Jump Instructions . . . . .	4
1.3.6	Interrupt Instructions . . . . .	4
1.4	Pins . . . . .	5
<b>2</b>	<b>DLX Processor</b>	<b>5</b>
2.1	History . . . . .	5
2.2	Registers . . . . .	5
2.3	Instruction Set . . . . .	5
2.3.1	Addressing Modes . . . . .	6
2.3.2	Data Transfer Instructions . . . . .	6
2.3.3	Arithmetic and Logical Instructions . . . . .	6
2.3.4	Control Transfer Instructions . . . . .	6
2.4	Pins . . . . .	6
<b>3</b>	<b>4stack</b>	<b>6</b>
3.1	History . . . . .	6
3.2	Registers . . . . .	7
3.3	Instruction Set . . . . .	7
3.3.1	Stack Operations . . . . .	8
3.3.2	Data Move Operations . . . . .	8
3.3.3	Flow Control Operations . . . . .	8
<b>4</b>	<b>Comparison</b>	<b>8</b>
<b>5</b>	<b>References</b>	<b>9</b>

# 1 MOS Technology 6502

## 1.1 History

The MOS Technology 6502 is an 8-bit microprocessor that was designed by Chuck Peddle for MOS Technology in 1975. It has approximately 5000 transistors.

The internal logic runs at the same speed as the external clock rate, but despite the slow clock speeds between 20 KHz and 2 MHz, the 6502 was actually competitive with other CPUs using significantly faster clocks. This is partly due to a simplistic state machine implemented by combinatorial logic to a greater extent than in many other designs; the two phase clock can thereby control the whole machine-cycle directly.

When the 6502 was introduced, it was the least expensive full-featured CPU on the market by a considerable margin, costing less than one-sixth the price of competing designs from larger companies such as Motorola and Intel.

His main concurrent was Zilog Z80.

6502 was the first microprocessor CPU with a 1-step instruction pipeline. This means that during execution of one command already the next instruction could be fetched.

One of the first uses for the design was the Apple I computer. The 6502 was next used in the Apple II, and the Commodore PET. It was later used in the Atari home computers, the BBC Micro family, and a huge number of other designs. Bender, a fictional android "industrial robot" and a main character in the animated TV series Futurama, was revealed to have a 6502 as his "brain".

## 1.2 Registers

6502 has very few registers. It has an 8-bit accumulator, two 8-bit index registers, one 8-bit stack pointer, an 8-bit status register, and a 16-bit program counter.

The accumulator is the main register for arithmetic and logic operations. Unlike the index registers X and Y, it has a direct connection to the Arithmetic and Logic Unit (ALU).

The stack memory ranges from 0x0100 to 0x01FF. The stack register (S) is a 8-bit offset to the stack page. In other words, whenever anything is being pushed on the stack, it will be stored to the address 0x0100+S.

The bits in this status register are called flags. The bits in the register are Negative, Overflow, Unused, Break, Decimal mode, Interrupt disable, Zero, and Carry flags.

## 1.3 Instruction Set

The 6502 microprocessor has a variable instruction encoding and the byte-order is Little Endian. Every command needs between 2 and 7 clock cycles. The instruction set includes 56 instructions.

### 1.3.1 Addressing Modes

The chip used the index and stack registers effectively with several addressing modes, including a fast "direct page" or "zero page" mode, that accessed memory locations from address 0 to 255 with a single 8-bit address. The 6502 has altogether 13 addressing modes.

The 6502 has a 64 KByte address space.

### 1.3.2 Arithmetic Instructions

Available commands are addition/subtraction, binary connections, and rotate/shift operations.

### 1.3.3 Compare Instructions

The compare instructions set or clear three of the status flags Carry, Zero, and Negative.

The three types of compare instructions are CMP (Compare Memory and Accumulator), CPX (Compare Memory and Index X), and CPY (Compare Memory and Index Y).

### 1.3.4 Register Instructions

These commands include load and store, transfer, flag operations and push/pull instructions.

### 1.3.5 Jump Instructions

The 6502 has some conditional and some unconditional jumps.

### 1.3.6 Interrupt Instructions

There are exact 4 instructions available: sei, cli, brk, nop.

## 1.4 Pins

The 6502 has a 16-bit address bus and a 8-bit data bus. As the memory was faster than the CPU, it made sense to optimize the CPU for memory access.

Altogether the 6502 microprocessor has 40 pins.

## 2 DLX Processor

### 2.1 History

The DLX processor is a RISC processor that was designed by John Hennessy and David Patterson with the main objective to produce a fully pipelined DLX processor for pedagogical purposes. It was first mentioned 1995 in "Computer Architecture: A Quantitative Approach." The pipeline of the DLX processor has 5 stages, namely fetch, decode, execute, memory, and writeback.

This processor is a load/store machine and emphasizes a simple instruction set, design for pipelining efficiency, an easily decoded instruction set, and efficiency as a compiler target.

### 2.2 Registers

The DLX processor has 32 general-purpose registers (GPRs), each of which is 32 bits long. Register r0 is a special register that always has the value 0.

Furthermore, the processor has 32 floating-point registers (FPRs), which can be used as 32 single precision 32-bit registers or as even-odd pairs holding double-precision values.

Last but not least, the DLX processor has a 32 bit program counter (PC) and 31 special purpose registers.

### 2.3 Instruction Set

The DLX processor has a hybrid instruction encoding. Each instruction is encoded in a 32-bit word. It uses a Big Endian addressing scheme. There are 3 instruction formats, namely I-type, R-type, and J-type. The I-type format is generally used for arithmetic and logic instructions that have an immediate operand, and for branch instructions. The R-type format is used for arithmetic and logic instructions that operate entirely on data in registers. The J-type instructions are used for unconditional jump instructions.

## 3 4stack

64 basic instructions are supported by the DLX processor, though it can also support extended instructions, as long as those instructions work purely on registers.

### 2.3.1 Addressing Modes

The DLX processor has only 3 addressing modes, namely immediate, displacement, and register.

The processor has a 4 GByte address space.

### 2.3.2 Data Transfer Instructions

The load and store instructions are the only means of transferring data between the CPU and memory.

### 2.3.3 Arithmetic and Logical Instructions

All ALU instructions are register-register instructions and contain addition, subtraction, multiplication, division, comparison, and so on.

### 2.3.4 Control Transfer Instructions

Control Transfer Instructions are mainly branches and jumps. There are also instructions available which deal with exceptions and interrupts.

## 2.4 Pins

The DLX processor has a 32 bit address and data bus. Together with clk, reset, and so on, the processor has 73 pins.

# 3 4stack

## 3.1 History

The 4stack processor is still a research project for high performance and low cost computing and designed by B. Paysan. The 4stack processor uses stack based instructions for a four way VLIW processor.

The 4stack processor has 4 Arithmetic Logic Units and 4 Stacks. Each stack has its own ALU. In addition to the 4 ALUs, two memory units allow parallel load and store.

### 3 4stack

The stacks store a 32 bit value and 64 bit data is represented by two stacks. Stack instructions either use 32 or 64 bit signed or unsigned integers or bit patterns, or 32 bit single or 64 bit double floats. Memory instructions load and store bytes, half words, words, and double words.

Less than 500k transistors are required for the core, leaving more space for caches. Furthermore, the stack paradigm greatly increases instruction density. The 4stack processor encodes up to 8 operations in 64 bits.

## 3.2 Registers

Each stack has 10 registers, including the stack pointer and the status register. There are 4 additional global registers for special purposes. For memory access there are further 32 registers.

## 3.3 Instruction Set

The 4stack processor is a 32 bit machine. The command length is 64 bits, though each command can consist of several operation fields for the independent execution units. All these operations are performed simultaneous.

The load instruction takes 2 cycles, though the store instruction takes only 1 cycle. In case of a cache miss a wait instruction has to be inserted.

The instruction encoding uses Big Endian and has 5 main instruction formats:

Normal Instructions	The normal instruction consists of four stack operations and two data move operations.
Conditional Setup Instructions	The conditional setup instruction consists of four stack operations and four corresponding conditional setup operations.
Branch Instructions	The branch instruction consists of four stack operations and one branch instruction.
Call Instructions	The call instruction consists of three stack operations and one instruction pointer-relative or absolute call or jump instruction.
Far Call Instruction	The far call instruction consists of one absolute call instruction.

## 4 Comparison

### 3.3.1 Stack Operations

Stack operations are divided into ALU operations and immediate number operations. The immediate number operations are intended to push small numbers on the stack. The ALU operations are used for general purpose and for floating point calculations.

### 3.3.2 Data Move Operations

Data move operations are divided into load/store, address update and immediate offset operations. An immediate offset operation in one data move field is added to the computed address in the other data move field.

### 3.3.3 Flow Control Operations

Flow control operations divide into conditional branches, calls/jumps, counted loops, returns, and indirect calls/jumps.

## 4 Comparison

	<b>6502</b>	<b>DLX</b>	<b>4stack</b>
Date of release	1975	1995	Under development
Architecture	Accumulator	RISC	VLIW and Stack
Internal data bus width	8	32	
External data bus width	8	8	
# of data registers	2 (X, Y)	30 GP, 32 FP	8 for each stack
# of other registers	1 accumulator, 1 program counter, 1 status register	31 special purpose	SP and status for each stack, 4 special purpose, 32 memory access
Instruction length	2-7 bytes	4 bytes	8 bytes
# of instructions	56	64	
Cycles per Instruction	1; 2 if a page boundary is crossed	4-5	
Pipeline	2 stage pipeline	5 stage pipeline	3 stage pipeline
Address bus width	16 bit	32 bit	32 or 64 bit
Endianness	Little-endian	Big-endian	Little-endian and Big-endian
Pins	40	73	
Exceptions	14	8	64
Interrupts	1 non-masked interrupt, 1 IRQ, and software interrupt	16	

## 5 References

- Hennessy, J.L. and Patterson, D.A. (1995): *Computer Architecture: A Quantitative Approach* Morgan Kaufmann
- Miller, Ethan (1996): *The DLX Processor*  
<http://www.csee.umbc.edu/courses/undergraduate/411/spring96/dlx.html>
- Naberezny, Mike (2001): *The 6502 microprocessor resource*  
<http://www.6502.org/tutorials>
- Paysan, Bernd (2000a): *A Four Stack Processor*  
<http://www.jwdt.com/%7Epaysan/4stack.pdf>
- Paysan, Bernd (2000b): *4stack Processor's User Manual*  
<http://www.jwdt.com/%7Epaysan/userman.pdf>
- Pickens, John and Clark, Bruce (2001): *NMOS 6502 Opcodes*  
<http://www.6502.org/tutorials/6502opcodes.html>
- Sonninen, Jarkko;Valta, Jouko; West, John and Mäkelä, Marko (1994): *Documentation for 6502* <http://www.nvg.ntnu.no/bbc/doc/6502.txt>