The CaNS 4-bit CPU

Martin Serpell

Abstract

The CaNS 4-bit CPU is a simple processor used to teach processor architecture in the Computer and Network Systems (UFCF93-30-1) module at the University of the West of England.

1. Introduction

The CaNS 4-bit CPU is a simple processor that has 16 instructions and can address 256 nibbles (4-bits) of memory.

2. Registers

The CaNS 4-bit CPU has four registers; a 4-bit accumulator, a 4-bit status register, an 8-bit program counter and an 8-bit stack pointer, as shown in Figure 1. All load, input, output and mathematical operations are carried out on 4-bit data variables and involve the Accumulator. The outcome of each operation is recorded within the Status register.

The Status register contains three Flags; single bit pieces of memory. The Zero (Z) Flag indicates that an operation has led to a variable (usually the Accumulator) having the value zero (Z=1). The Negative (N) Flag indicates that an operation has created a variable with a negative value, i.e. the most significant bit (msb) is set to 1. The Carry (C) Flag indicates that there has been an overflow or a borrow during an addition or subtraction (C=1).

The Program Counter points to the next instruction to be executed. This register is also called the Instruction Pointer.

The Stack Pointer points into memory where a stack is maintained. The stack is a data structure that is used to save memory addresses and variables. It is used by the Jump to Subroutine (JSR) instruction and the Return from Subroutine (RET) instruction. It is also used by the interrupt mechanism and the Return from Interrupt (RTI) instruction.

Table 1 contains a description of the CaNS 4-bit CPU registers.

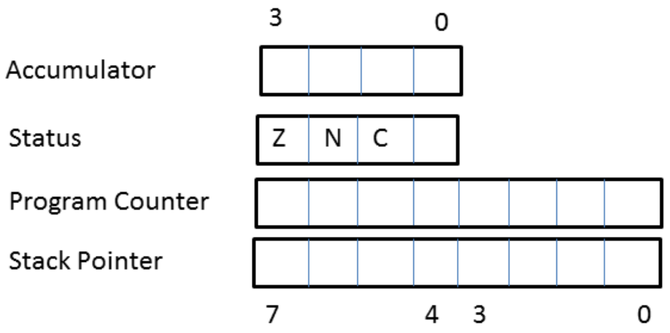

Figure 1: The CaNS 4-bit CPU Registers

3. Instructions

The CaNS 4-bit CPU has the following sixteen instructions.

NOP No operation

LD data Load immediate. The data immediately follows the op-code in program memory. The data is loaded into the Accumulator.

LD address Load direct. The address of the data follows the op-code in program memory; in this case high nibble followed by low nibble. The data is loaded into the Accumulator.

ST address Store direct. The address of the data follows the op-code in program memory; in this case high nibble followed by low nibble. The data is written from the Accumulator into memory.

ADD address Add data located at address to the contents of the Accumulator.

SUB address Subtract data located at address from the contents of the Accumulator.

CMP address Compare the value of the data located at address to the value of the Accumulator.

JMP address Jump to address.

JZ address Jump to address if flag Z = 1.

JN address Jump to address if flag N = 1.

JC address Jump to address if flag C = 1.

JSR address Call a subroutine at address.

RET Return from subroutine.

IN port Read from a port. The port number immediately follows the opcode in program memory. The data is loaded into the Accumulator.

OUT port Write to a port. The port number immediately follows the opcode in program memory. The data is written from the Accumulator.

RTI Return from interrupt.

Tables 2 and 3 contain descriptions of the CaNS 4-bit CPU instructions.

4. Interrupts

The CaNS 4-bit CPU has a single interrupt. On interrupt the contents of the Program Counter, Accumulator and Status Register are placed on the stack. The Program Counter is then loaded with the value 0xFC and the instruction at that location is then fetched and executed. By default a RTI instruction is placed there but this can be replaced with other code. The RTI instruction restores the Status Register, Accumulator and then the Program Counter from the stack. The next instruction pointed to by the Program Counter is then executed; this should be the instruction that should have been executed next if no interrupt had taken place.

5. Memory Layout

The program, data and stack all must reside within 256 nibbles of memory. Programs are loaded from location 0x00. Code for the interrupt is located at location 0xFC; on power up this is initialised to 0xF, the RTI instruction. On initialisation at power-up the Program Counter is set to 0x00 and the Stack Pointer to 0xFB. The stack starts in high memory and grows into low memory whereas the program is loaded in low memory and grows into high memory. It is important that the stack should never overflow into data and program. Table 4 shows the layout of the memory as used by the CaNS 4-bit CPU.

Register	**Number of Bits**	**Purpose**
Accumulator	4	Used to carry out operations on 4-bit data.
Status	4	Used to record the outcome of operations on 4-bit data. This register is often called Flags.
Program Counter	8	Points to the next instruction to be executed. This register is also called the Instruction Pointer.
Stack Pointer	8	Points into memory where a stack is maintained.

Table 1: The CaNS 4-bit CPU Registers

Op-code	**Mneumonic**	**Number of Nibbles**	**Description**	**Flags**		
				Z	N	C
0	NOP	1	No operation	-	-	-
1	LD	2	Load	*	*	0
2	LD	3	Load	*	*	0
3	ST	3	Store	-	-	-
4	ADD	3	Add	*	*	*
5	SUB	3	Subtract	*	*	*
6	CMP	3	Compare	*	*	*
7	JMP	3	Jump	-	-	-
8	JZ	3	Jump on Z	-	-	-
9	JN	3	Jump on N	-	-	-
A	JC	3	Jump on C	-	-	-
B	JSR	3	Jump to Subroutine	-	-	-
C	RET	1	Return	-	-	-
D	IN	2	Input	*	*	0
E	OUT	2	Output	-	-	-
F	RTI	1	Return from interrupt	*	*	*

Table 2: The CaNS 4-bit CPU Instructions; - indicates flag is unchanged, 0 that the flag is cleared, and * the flag changed by the instruction

Mneumonic	Instruction Nibbles
NOP	0
LD	1 Data
LD	2 High Address Nibble Low Address Nibble
ST	3 High Address Nibble Low Address Nibble
ADD	4 High Address Nibble Low Address Nibble
SUB	5 High Address Nibble Low Address Nibble
CMP	6 High Address Nibble Low Address Nibble
JMP	7 High Address Nibble Low Address Nibble
JZ	8 High Address Nibble Low Address Nibble
JN	9 High Address Nibble Low Address Nibble
JC	A High Address Nibble Low Address Nibble
JSR	B High Address Nibble Low Address Nibble
RET	C
IN	D Port
OUT	E Port
RTI	F

Table 3: The CaNS 4-bit CPU Instruction formats

Addresses	**Usage**
00..XX	Program and data
XX..FB	Stack
FC..FF	Interrupt code

Table 4: The CaNS 4-bit Memory Locations; XX is a value between 00 and FB