D16 Processor Reference Manual

Michael Nolan

Aug 16, 2016

Contents

1 The Processor

The D16 Processor is a very simple, RISC like 16 bit processor with variable length instructions. It has 8 general purpose registers, 32 special purpose registers, and support for up to 64K of memory.

1.1 General Purpose Registers

The D16 processor defines 8 general purpose registers, called r0 - r7. 2 of these, although they behave the same as the other registers, have special meaning to the processor and in the ABI, and they are as follows:

r6: This is generally used as the pointer to the start of a stack frame, but has no special meaning to the processor

r7: This is the stack pointer, and is manipulated via the stack instructions (push and pop)

1.2 Flags

The processor also contains several flags in Special Register 0.

Zero	set if the result of the last computation is 0
Sign	set if the result is negative (bit 15 is set)
Carry	set if there was a carry or borrow in the past computation
oVerflow	set if there was a signed overflow in the last computation

2 Instruction Set

Most instructions come in 2 formats, register and immediate. The immediate versions of an instruction will have bit 7 set in the opcode field and the 16 bit immediate in the word following the instruction. In the subsequent definitions, op2 will refer to the immediate value if the instruction has an immediate, otherwise it refers to rS.

2.1 ADD

Immediate	opcode	Unused	source	dest
Imm	000001	00	rS	rD

```
ADD <rD>, <rS or immediate>
```

rD = rD + op2
Updates flags

2.2 SUB

Immediate	opcode	Unused	source	dest
Imm	000010	00	rS	rD

```
SUB <rD>, <rS or immediate>
```

rD = rD - op2
Updates flags

2.3 PUSH

Immediate	opcode	Unused	source	dest
Imm	000011	00	000	rD

```
PUSH <rD or immediate>
```

r7 = r7 - 2
memory[r7] = rD
This instruction does not update the flags

2.4 POP

Immediate	opcode	Unused	source	dest
Imm	000100	00	000	rD

```
POP <rD or immediate>
```

rD = memory[r7]
r7 = r7 + 2
Does not update flags

2.5 MOV

Mov has 2 different encodings depending whether the immediate (if any) fits into 1 byte.
Neither encoding updates the flags.

2.5.1 general MOV encoding

Immediate	opcode	Unused	source	dest
Imm	001101	00	rS	rD

```
MOV <rD>, <rS or immediate>
```

rD = op2
This encoding is used for register to register MOVs or when the immediate value

will not fit in one byte.

2.5.2 special byte MOV

Unused	opcode	data
0	000101 + rD	byte immediate

```
MOV <rD>, <byte immediate>
```

rD = immediate
This encoding is only used when the immediate will fit in 1 byte

2.6 AND

Immediate	opcode	Unused	source	dest
Imm	001110	00	rS	rD

```
AND <rD>, <rS or immediate>
```

rD = rD AND op2
This instruction updates the flags, and will reset the overflow and carry flags.

2.7 OR

Immediate	opcode	Unused	source	dest
Imm	001111	00	rS	rD

```
OR <rD>, <rS or immediate>
```

rD = rD OR op2
This instruction updates the flags, and will reset the overflow and carry flags.