Contents

[**Format Summary** 2](#_Toc143865380)

[**Microcode Summary** 3](#_Toc143865381)

[ALU Opcode Summary 3](#_Toc143865382)

[Load Operation Summary 3](#_Toc143865383)

[Jump code Summary 4](#_Toc143865384)

[**Macrocode Summary** 4](#_Toc143865385)

# **Bit16 Specifications**

## Registers

|  |  |  |
| --- | --- | --- |
| Code | Mnemonic | Purpose |
| 0 | A | General Purpose |
| 1 | B |
| 2 | C |
| 3 | D |
| 4 | E |
| 5 | SP | Stack pointer |
| 6 | LR | Link register |
| 7 | PC | Program counter |

## ROM

## RAM

# **Format Summary**

The bit16 instruction set formats are shown in the following figure.



**Figure 1: bit16 instruction set formats**

# **Microcode Summary**

The following section summarizes the bit16 instruction set microcode.

## ALU Opcode Summary

The following table summarizes the bit16 instruction set operations.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Op3 code | Op4 code | Mnemonic | Instruction | Condition codes set | See Section: |
| 0 | 0 | ADD | Add |  |  |
|  | 1 | CMN | Compare negative | a |  |
| 1 | 2 | SUB | subtract |  |  |
|  | 3 | CMP | Compare | a |  |
| 2 | 4 | MUL | Multiply |  |  |
|  | 0 (5) | NOT | Bitwise not |  |  |
| 3 | 6 | AND | Bitwise and |  |  |
|  | 7 | TST | Test bits | a |  |
| 4 | 8 | OR | Bitwise or |  |  |
| 5 | 10 | XOR | Bitwise exclusive or |  |  |
|  | 11 | TEQ | Test bits equality | a |  |
| 6 | 12 | SHR | Shift right |  |  |
|  | 1 (13) | NEG | Negate |  |  |
| 7 | 14 | SHL | Shift left |  |  |
|  | 15 | MOV | Move to register |  |  |

**Table 1: bit16 instruction set opcodes**

## Load Operation Summary

The following table summarizes the Load operation for the bit16 instruction set.

|  |  |  |
| --- | --- | --- |
| Mnemonic | Instruction | See Section: |
| LD | Load/store |  |

**Table 2: bit16 load instruction**

## Jump code Summary

The following table summarizes the jump codes for the bit16 instruction set.

|  |  |  |  |
| --- | --- | --- | --- |
| Jump code | Mnemonic | Instruction | See Section: |
| 0 | JMP | **J**u**mp** |  |
| 1 | JEQ | **J**ump if **eq**ual |  |
| 2 | JNE | **J**ump if **n**ot **e**qual |  |
| 3 | JGT | **J**ump if **g**reater **t**han |  |
| 4 | JLT | **J**ump if **l**ess **t**han |  |
| 5 | JGE | **J**ump if **g**reater than or **e**qual to |  |
| 6 | JLE | **J**ump if **l**ess than or **e**qual to |  |
| 7 | JNV | **J**ump **n**e**v**er |  |

**Table 3: bit16 jump instructions**

# **Macrocode Summary**

The following section summarizes the bit16 instruction set macrocode. Macrocode instructions are instructions that are not supported by the hardware and are made up of one or more microcode instructions.

|  |  |  |
| --- | --- | --- |
| Mnemonic | Instruction | Microcode |
| PSH | Push register(s) to stack | PSH A = SUB SP, 1  LD [SP], A |
| POP | Pop register(s) from stack | POP A = ADD SP, 1  LD A, [SP, -1] |
| CALL | Call procedure | CALL L = MOV LR, PC  ADD LR, 1  JMP L |
| RET | Return from procedure | RET = MOV PC, LR |

**Table 4: bit16 macrocode instructions**

# **Formats**

## Format 0: No Operation



**Figure 2: Format 0**

## Format 1: ALU Operation



A: Source/destination register

B: Source Register2

Op: Opcode

**Figure 3: Format 1**

## Format 2: ALU Operation with Constant



## Format 3: Binary ALU Operation with Register



## Format 4: Binary ALU Operation with Constant



## Format 5: Unary ALU Operation



## Format 6: Load





## Format 7: Jump

