AmigaFlight Bit Manipulation Instructions

Andrew Duffy Morris

COLLABORATORS			
	TITLE : AmigaFlight Bit Manipulation Instructions		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Andrew Duffy Morris	July 20, 2024	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

AmigaFlight Bit Manipulation Instructions

1.1 AmigaFlight® Help: Bit Manipulation Instructions

```
Bit Manipulation
================

Bit manipulation operations are accomplished using the following
instructions.


  Bit Testing
  -----------
  BTST              Test a Bit

  Bit Setting
  -----------
  BSET              Test a Bit and Set

  Bit Clearing
  ------------
  BCLR              Test a Bit and Clear

  Bit Changing
  ------------
  BCHG              Test a Bit and Change
```

1.2 AmigaFlight® Help: Test a Bit

```
BTST  Test a Bit
================

  Place the value of the specified bit, from the destination
  address, in the Z condition code. If the specified destination
  is a data register, then the specified bit offset is modulo 32.
  If the specified destination address is a memory location, then
  the specified bit offset is modulo 8, and then the offset is
  applied to the byte location.
```

Complement (<bit no.> Destn) -> Z

Assembler Syntax

```
BTST{.[B/L]}  Dn,<ea>
BTST{.[B/L]}  #<data>,<ea>
```

Dn or #<data> specify the bit no. and <ea> may be data only

Addressing Modes

Mode	Source	Destination
Data Register Direct	-	*
Address Register Direct	-	-
Address Register Indirect	-	*
Postincrement Register Indirect	-	*
Predecrement Register Indirect	-	*
Register Indirect with Offset	-	*
Register Indirect with Index	-	*
Absolute Short	-	*
Absolute Long	-	*
P.C. Relative with Offset	-	-
P.C. Relative with Index	-	-
Immediate	-	-

Data Size

Byte, Long

If <ea> is Data register then
#<data> 0 - 31 else #<data> 0 - 7

Status Flags

N Not affected
Z Set if bit tested = 0, else clear
V Not affected
C Not affected
X Not affected

Instruction Size and Cycles to Execute

<ea>	Dn,<ea>		#<data>,<ea>	
	#	p	#	p
Dn	2	6	4	10
(An)	2	8	4	12
(An)+	2	8	4	12
-(An)	2	10	4	14
d16(An)	4	12	6	16
d8(An,Ri)	4	14	6	18

```
  Abs short 4   12    6    16
  Abs long  6   16    8    20
  d16(PC)   4   12    6    16
  d8(PC,Ri) 4   14    6    18

  # = no. of instruction bytes
  p = no. of instruction clock periods
```

1.3 AmigaFlight® Help: Test a Bit and Set

```
BSET  Test a Bit and Set
========================

  Place the value of the specified bit, from the destination
  address, in the Z condition code, and then set the specified
  bit in the destination address to a one. If the specified
  destination is a data register, then the specified bit offset is
  modulo 32. If the specified destination address is a memory
  location, then the specified bit offset is modulo 8, and then the
  offset is applied to the byte location.

  Complement of (<bit no.> Destn) -> Z

  1 -> (<bit no.> Destn)


Assembler Syntax
----------------
  BSET{.[B/L]}  Dn,<ea>
  BSET{.[B/L]}  #<data>,<ea>

  Dn or #<data> specify the bit no. and <ea> may be data alterable
  only


Addressing Modes
----------------
  Mode              Source  Destination

  Data Register Direct      - *
  Address Register Direct     - -
  Address Register Indirect   - *
  Postincrement Register Indirect   - *
  Predecrement Register Indirect    - *
  Register Indirect with Offset   - *
  Register Indirect with Index    - *
  Absolute Short        - *
  Absolute Long       - *
  P.C. Relative with Offset   - -
  P.C. Relative with Index    - -
  Immediate       - -


Data Size
---------
```

```
  Byte, Long

  If <ea> is Data register then #<data> 0 - 31 else #<data> 0 - 7


Status Flags
------------
  N  Not affected
  Z  Set if bit tested = 0, else clear
  V  Not affected
  C  Not affected
  X  Not affected


Instruction Size and Cycles to Execute
--------------------------------------
  <ea>     Dn,<ea>    #<data>,<ea>
      #     p     #      p
  Dn     2    <8     4    <12
  (An)     2   12     4    16
  (An)+    2   12     4    16
  -(An)    2   14     4    18
  d16(An)    4   16     6    20
  d8(An,Ri)  4   18     6    22
  Abs short  4   16     6    20
  Abs long   6   20     8    24

  # = no. of instruction bytes
  p = no. of instruction clock periods
```

1.4 AmigaFlight® Help: Test a Bit and Clear

```
BCLR  Test a Bit and Clear
==========================

  Place  the  value  of  the  specified  bit,~ from  the destination
  address, in the Z condition code, and then clear the specified bit
  in the destination address to a zero. If the specified destination
  is a data register, then the specified bit  offset is  modulo  32.
  If the  specified destination  address  is a memory location, then
  the specified  bit offset  is modulo 8, and  then  the  offset  is
  applied to the byte location.

  Complement of (<bit no.> Destn) -> Z

  0 -> (<bit no.> Destn)


Assembler Syntax
----------------
  BCLR{.[B/L]}  Dn,<ea>
  BCLR{.[B/L]}  #<data>,<ea>

  Dn or #<data> specify the  bit no. and <ea> may  be data alterable
  only
```

```
Addressing Modes
----------------
  Mode              Source  Destination

  Data Register Direct      - *
  Address Register Direct     - -
  Address Register Indirect   - *
  Postincrement Register Indirect   - *
  Predecrement Register Indirect    - *
  Register Indirect with Offset   - *
  Register Indirect with Index    - *
  Absolute Short        - *
  Absolute Long       - *
  P.C. Relative with Offset   - -
  P.C. Relative with Index    - -
  Immediate       - -


Data Size
---------
  Byte, Long

  If <ea> is Data register then #<data> 0 - 31 else #<data> 0 - 7


Status Flags
------------
  N  Not affected
  Z  Set if bit tested = 0, else clear
  V  Not affected
  C  Not affected
  X  Not affected


Instruction Size and Cycles to Execute
--------------------------------------
  <ea>    Dn,<ea>   #<data>,<ea>
      #    p    #     p
  Dn    2   <8    4   <12
  (An)    2   12    4    16
  (An)+   2   12    4    16
  -(An)   2   14    4    18
  d16(An)   4   16    6    20
  d8(An,Ri) 4   18    6    22
  Abs short 4   16    6    20
  Abs long  6   20    8    24

  # = no. of instruction bytes
  p = no. of instruction clock periods
```

1.5 AmigaFlight® Help: Test a Bit and Change

BCHG Test a Bit and Change
==========

  Place the value of the specified bit, from the destination address, in the Z condition code, and then complement the specified bit in the destination address. If the specified destination is a data register, then the specified bit offset is modulo 32. If the specified destination address is a memory location, then the specified bit offset is modulo 8, and then the offset is applied to the byte location.

  Complement of (<bit no.> Destn) -> Z
  Complement of (<bit no.> Destn) -> (<bit no.> Destn)

Assembler Syntax

  BCHG{.[B/L]} Dn,<ea>
  BCHG{.[B/L]} #<data>,<ea>

  Dn or #<data> specify the bit no.
  and <ea> may be data alterable only

Addressing Modes

  Mode Source Destination

  Data Register Direct - *
  Address Register Direct - -
  Address Register Indirect - *
  Postincrement Register Indirect - *
  Predecrement Register Indirect - *
  Register Indirect with Offset - *
  Register Indirect with Index - *
  Absolute Short - *
  Absolute Long - *
  P.C. Relative with Offset - -
  P.C. Relative with Index - -
  Immediate - -

Data Size

  Byte, Long

  If <ea> is Data register then
  #<data> 0 - 31 else #<data> 0 - 7

Status Flags

  N Not affected
  Z Set if bit tested = 0, else clear
  V Not affected
  C Not affected

```
  X  Not affected


Instruction Size and Cycles to Execute
--------------------------------------
  <ea>     Dn,<ea>    #<data>,<ea>
      #     p     #      p
  Dn     2    <8     4    <12
  (An)     2   12     4     16
  (An)+    2   12     4     16
  -(An)    2   14     4     18
  d16(An)    4   16     6     20
  d8(An,Ri)  4   18     6     22
  Abs short  4   16     6     20
  Abs long   6   20     8     24

  # = no. of instruction bytes
  p = no. of instruction clock periods
```