**1)**

**1110 001 0 0000 0001 0000 0000 00000000**

Cond = 1110 = always

Bit[27:25] = 001 = Data Processing Immediate

Opcode = 0000 = AND

S = 0, no altera condition codes

Rn = R1

Rd = R0

*32-bit immediate shifter operand*

2)

**1110 001 1100 0 0000 0001 0000 00101000**

Cond = 1110 = always

Bit[27:25]= 001 = Data Processing Immediate

Opcode = ORR = 1100

S = 0, no altera cond codes

Rn = R0

Rd = R1

*32-bit immediate shifter operand*

3)

**1110 011 11101 0001 0010 00000000 0000**

Cond = 1110 = always

Bit[27:25] = 011 = Load/Store register offset

Addressing Mode 2

P = 1, U=1, B=1, W=0, L=1

Suma,byte,Load

Rn = R1

Rd = R2

Rm = R0

*Addressing Mode 2*

4)

**1110 010 11101 0001 0011 000000000010**

Cond = always

Bits[27:25] = 010 = Load/Store Immediate Offset(12-bits)

P=1, U=1, B=1, W=0, L=1

Suma,Byte,Load

Rn = R1

Rd = R3

*Addressing Mode 2*

**5)**

**1110 000 0100 0 0000 0101 00000 00 0 0000**

Cond = always

Bits[27:25] = 000 = Data Processing Immediate Shift

Opcode = 0100 = *ADD*

S = 0, cond codes no se alteran

Rn = R0

Rd = R5

Shift\_imm = 0

Shift = 00 = LSL

Rm = R0

6)

**1110 000 0100 0 0010 0101 00000 00 0 0101**

Cond = 1110 = always

Bits[27:25] = 000 = Data Processing Immediate Shift

Opcode = 0100 = *ADD*

S = 0, cond codes no se alteran

Rn = R2

Rd = R5

Shift\_imm = 0

Shift = 00 = LSL

Rm = R5

7)

**1110 001 0010 1 0011 0011 0000 00000001**

Cond = 1110 = always

Bits[27:25] = 001 = Data processing immediate

Opcode = 0010 = *SUB*

S = 1, cond codes can be modified

Rn = R3

Rd = R3

Rotate\_imm = 0

Immed\_8 = 1

8)

**0001 101 0 111111111111111111111101**

Cond = 0001 = NE(Not Equal)

Bits[27:25] = 101 = Branch & branch with link

L = 0 = *Branch*

9)

**1110 010 11100 0001 0101 000000000011**

Cond = 1110 = always

Bits[27:25]= 010 = Load/Store Immediate Offset

P =1, U=1, B=1, W=0, L=0

Suma,byte,store

Rn = R1

Rd = R5

Offset = 000000000011

*Addressing Mode 2*

10)

**1110 101 0 000000000000000000000001**

Cond = always

Bits[27:25] = 101 = branch & branch with link

L = 0 = *Branch*

11)

**0000 101 1 000001010000011100000100**

Cond = 0000 = EQ(equal)

Bits[27:25] = 101 = branch & branch with link

L = 1 = *Branch with Link*

12)

**1110 101 0 111111111111111111111111**

Cond = 1110 = always

Bits[27:25] = 101 = branch & branch with link

L = 0 = *branch*