```
*********************************************
* LOAD AND STORE INSTRUCTIONS   [op-code : 00] *
*********************************************

15___13_____10______7_____5_______2___0_
|0|0| |i|n|s| |d|s|t| |a|dd| |s|r|c1| |s|r|c2|          [LOAD]
-----------------------------------------

ins     : Indicator whether the it is a Load or Store instruction
dst     : Destination operand
add     : Addressing Mode for the 2nd operand
src1    : Source operand 1
src2    : Source operand 2

15___13_____10______7____5_______2___0_
|0|0| |0|1|0| |s|r|c| |a|dd| |d|s|t1| |d|s|t2|          [STORE]
-----------------------------------------

ins     : Indicator whether the it is a Load or Store instruction
src     : Source Register
add     : Addressing Mode for the 2nd operand
dst1    : Destination operand 1
dst2    : Destination operand 2


>>Load Immediate                          (li)
=============================================

15___13_____10______7___5___2_0_
|0|0| |0|0|1| |d|s|t| |0|0| |||| ||||
-------------------------------------
_____________________________________
|             IMMEDIATE             |
-------------------------------------

li Ra, #100

MAB <- PC       (tbP<-1,wMAB<-1)
MDB <- M[MAB]   (wMDB<-1)
Ra <- MDB       (tbMDB<-1, tbB<-1, wD<-1)
PC <- PC+1      (tbP<-1, P/M<-1, tbM<-1, wPC<-1)


>>Load Register                           (lr)
=============================================


15___13_____10______7____5_______2_0_
|0|0| |0|0|1| |d|s|t| |0|1| |s|r|c1| ||||
----------------------------------------

lr Ra, Rb

R <- Rb (tbD2<-1, wR<-1)
Ra <- R (tbR<-1, wD<-1)




>>Load with Base Indexed Addressing     (lx)
=============================================
```

```
15____13______10_______7_____5________2____0_
|0|0| |0|0|1| |d|s|t| |1|0| |s|r|c1| |s|r|c2|
-----------------------------------------------

_________________________________________________
|                IMMEDIATE                      |
-------------------------------------------------

lx Ra, 10( Rb, Rc)

MAB <- PC        (tbP<-1,wMAB<-1)
MDB <- M[MAB]    (wMDB<-1)
R <- MDB         (tbDB<-1, wR<-1)
T <- R + Rb      (tbD2<-1, tbA<-1, wT<-1)
R <- Rc          (tbD3<-1, wR<-1)
MAB <- T + R     (tbT<-1, tbA<-1, wMAB<-1)
MDB <- M[MAB]    (wMDB<-1)
Ra <- MDB        (tbMDB<-1, tbB<-1, wD<-1)
PC <- PC+1       (tbP<-1, P/M<-1, tbM<-1, wPC<-1)


>>Load with Memory Indirect Addressing  (ldn)
=============================================


15____13______10_______7_____5________2____0_
|0|0| |0|0|1| |d|s|t| |1|1| |s|r|c1| |s|r|c2|
-----------------------------------------------

_________________________________________________
|                IMMEDIATE                      |
-------------------------------------------------

ldn Ra, @10( Rb, Rc)

MAB <- PC        (tbP<-1,wMAB<-1)
MDB <- M[MAB]    (wMDB<-1)
R <- MDB         (tbDB<-1, wR<-1)
T <- R + Rb      (tbD2<-1, tbA<-1, wT<-1)
R <- Rc          (tbD3<-1, wR<-1)
MAB <- T + R     (tbT<-1, tbA<-1, wMAB<-1)
MDB <- M[MAB]    (wMDB<-1)
MAB <- MDB       (tbMDB<-1, tbB<-1, wMAB<-1)
MDB <- M[MAB]    (wMDB<-1)
Ra <- MDB        (tbMDB<-1, tbB<-1, wD<-1)
PC <- PC+1       (tbP<-1, P/M<-1, tbM<-1, wPC<-1)



>>Store with Base Indexed Addressing    (stx)
=============================================


15____13______10_______7_____5________2____0_
|0|0| |0|1|0| |s|r|c| |1|0| |d|s|t1| |d|s|t2|
-----------------------------------------------

_________________________________________________
|                IMMEDIATE                      |
-------------------------------------------------

stx -5(Rb), Rc

MAB <- PC        (tbP<-1,wMAB<-1)
MDB <- M[MAB]    (wMDB<-1)
R <- MDB         (tbDB<-1, wR<-1)
MAB <- R + Rb    (tbA<-1, wMAB<-1, tbD2<-1)
MDB <- Rc        (tbD3<-1, xMDB<-01, wMDB<-1)
```

```
PC <- PC+1      (tbP<-1, P/M<-1, tbM<-1, wPC<-1)


>>Store with Memory Indirect Addressing (sdn)
============================================


15____13_____10_______7_____5________2____0_
|0|0| |0|1|0| |s|r|c| |1|1| |d|s|t1| |d|s|t2|
---------------------------------------------
_____________________________________________
|               IMMEDIATE                   |
---------------------------------------------

sdn @-5(Rb), Rc

MAB <- PC       (tbP<-1,wMAB<-1)
MDB <- M[MAB]   (wMDB<-1)
R <- MDB        (tbDB<-1, wR<-1)
MAB <- R + Rb   (tbA<-1, wMAB<-1, tbD2<-1)
MDB <- M[MAB]   (wMDB<-1)
MAB <- MDB      (tbMDB<-1, tbB<-1, wMAB<-1)
MDB <- Rc       (tbD3<-1, xMDB<-01, wMDB<-1)
PC <- PC+1      (tbP<-1, P/M<-1, tbM<-1, wPC<-1)


*********************************************************
* ARITHMETIC AND LOGICAL INSTRUCTIONS   [op-code : 01] *
*********************************************************



15____13_____10_______7_____5________2____0_
|0|1| |a|l|u| |d|s|t| |a|dd| |s|r|c1| |s|r|c2|
---------------------------------------------

ins     : Signifies the ALU Operation
dst     : Destination operand
add     : Addressing Mode for the 2nd operand
src1    : Source operand 1
src2    : Source operand 2



>>Add Immediate                          (addi)
==============================================


15____13_____10_______7____5____2_0_
|0|1| |0|0|0| |d|s|t| |0|0| |||| ||||
-------------------------------------
_____________________________________
|             IMMEDIATE             |
-------------------------------------

addi Ra, #5

R <- Ra         (tbD2<-1, wR<-1)
MAB <- PC       (tbP<-1,wMAB<-1)
MDB <- M[MAB]   (wMDB<-1)
Ra <- MDB + R   (tbMDB <- 1, tbA<-1, wD<-1)
```

```
>>Add Register                              (addr)
==============================================


15____13______10_______7_____5______2_0_
|0|1| |0|0|0| |d|s|t| |0|1| |s|r|c| ||||
----------------------------------------

addr Ra, Rb

R <- Ra          (tbD2<-1, wR<-1)
Ra <- R+Rb       (tbD3<-1, tbA<-1, wD<-1)



>>Add with Base Indexed Addressing      (addx)
================================================


15____13______10_______7_____5________2___0__
|0|1| |0|0|0| |d|s|t| |1|0| |s|r|c1| |s|r|c2|
-------------------------------------------
_____________________________________________
|                IMMEDIATE                   |
---------------------------------------------

addx Ra, 10( Rb, Rc)

MAB <- PC        (tbP<-1,wMAB<-1)
MDB <- M[MAB]    (wMDB<-1)
R <- MDB         (tbDB<-1, wR<-1)
T <- R + Rb      (tbD2<-1, tbA<-1, wT<-1)
R <- Rc          (tbD3<-1, wR<-1)
MAB <- T + R     (tbT<-1, tbA<-1, wMAB<-1)
MDB <- M[MAB]    (wMDB<-1)
R <- Ra          (tbD2<-1, wR<-1)
Ra <- R + MDB    (tbMDB<-1, tbA<-1, wD<-1)
PC <- PC+1       (tbP<-1, P/M<-1, tbM<-1, wPC<-1)


>>Add with Memory Indirect Addressing   (addn)
================================================


15____13______10_______7_____5________2____0_
|0|1| |0|0|0| |d|s|t| |1|1| |s|r|c1| |s|r|c2|
-------------------------------------------
_____________________________________________
|                IMMEDIATE                   |
---------------------------------------------

addn Ra, @10( Rb, Rc)

MAB <- PC        (tbP<-1,wMAB<-1)
MDB <- M[MAB]    (wMDB<-1)
R <- MDB         (tbDB<-1, wR<-1)
T <- R + Rb      (tbD2<-1, tbA<-1, wT<-1)
R <- Rc          (tbD3<-1, wR<-1)
MAB <- T + R     (tbT<-1, tbA<-1, wMAB<-1)
MDB <- M[MAB]    (wMDB<-1)
MAB <- MDB       (tbMDB<-1, tbB<-1, wMAB<-1)
MDB <- M[MAB]    (wMDB<-1)
R <- Ra          (tbD2<-1, wR<-1)
Ra <- R + MDB    (tbMDB<-1, tbA<-1, wD<-1)
```

```
PC <- PC+1      (tbP<-1, P/M<-1, tbM<-1, wPC<-1)


>>Subtract Immediate                    (subi)
==========================================


15____13_____10_______7____5____2_0_
|0|1| |0|0|1| |d|s|t| |0|0| |||| ||||
-------------------------------------

_____________________________________
|             IMMEDIATE             |
-------------------------------------

subi Ra, #5

R <- Ra         (tbD2<-1, wR<-1)
MAB <- PC       (tbP<-1,wMAB<-1)
MDB <- M[MAB]   (wMDB<-1)
Ra <- MDB - R   (tbMDB <- 1, tbA<-1, wD<-1)


>>Subtract Register                     (subr)
==========================================


15____13_____10_______7_____5_____2_0_
|0|1| |0|0|1| |d|s|t| |0|1| |s|r|c| ||||
----------------------------------------

subr Ra, Rb

R <- Ra         (tbD2<-1, wR<-1)
Ra <- R - Rb    (tbD3<-1, tbA<-1, wD<-1)


>>Subtract with Base Indexed Addressing (subx)
==============================================


15____13_____10_______7_____5________2___0__
|0|1| |0|0|1| |d|s|t| |1|0| |s|r|c1| |s|r|c2|
-----------------------------------------------

_______________________________________________
|               IMMEDIATE                     |
-----------------------------------------------

subx Ra, 10( Rb, Rc)

MAB <- PC       (tbP<-1,wMAB<-1)
MDB <- M[MAB]   (wMDB<-1)
R <- MDB        (tbDB<-1, wR<-1)
T <- R + Rb     (tbD2<-1, tbA<-1, wT<-1)
R <- Rc         (tbD3<-1, wR<-1)
MAB <- T + R    (tbT<-1, tbA<-1, wMAB<-1)
MDB <- M[MAB]   (wMDB<-1)
R <- Ra         (tbD2<-1, wR<-1)
Ra <- R - MDB   (tbMDB<-1, tbA<-1, wD<-1)
PC <- PC+1      (tbP<-1, P/M<-1, tbM<-1, wPC<-1)


>>Subtract with Memory Indirect Addressing (subn)
================================================
```

```
15____13_____10_______7_____5________2____0_
|0|1| |0|0|1| |d|s|t| |1|1| |s|r|c1| |s|r|c2|
-------------------------------------------

_____________________________________________
|               IMMEDIATE                   |
---------------------------------------------

subn Ra, @10( Rb, Rc)

MAB <- PC       (tbP<-1,wMAB<-1)
MDB <- M[MAB]   (wMDB<-1)
R <- MDB        (tbDB<-1, wR<-1)
T <- R + Rb     (tbD2<-1, tbA<-1, wT<-1)
R <- Rc         (tbD3<-1, wR<-1)
MAB <- T + R    (tbT<-1, tbA<-1, wMAB<-1)
MDB <- M[MAB]   (wMDB<-1)
MAB <- MDB      (tbMDB<-1, tbB<-1, wMAB<-1)
MDB <- M[MAB]   (wMDB<-1)
R <- Ra         (tbD2<-1, wR<-1)
Ra <- R - MDB   (tbMDB<-1, tbA<-1, wD<-1)
PC <- PC+1      (tbP<-1, P/M<-1, tbM<-1, wPC<-1)



>>AND Immediate                          (andi)
=============================================


15____13_____10_______7____5____2_0_
|0|1| |0|1|0| |d|s|t| |0|0| |||| ||||
-------------------------------------

_____________________________________
|             IMMEDIATE             |
-------------------------------------

andi Ra, #5

R <- Ra         (tbD2<-1, wR<-1)
MAB <- PC       (tbP<-1,wMAB<-1)
MDB <- M[MAB]   (wMDB<-1)
Ra <- MDB & R   (tbMDB <- 1, tbA<-1, wD<-1)


>>AND Register                           (andr)
=============================================


15____13_____10_______7_____5_____2_0_
|0|1| |0|1|0| |d|s|t| |0|1| |s|r|c| ||||
----------------------------------------

andr Ra, Rb

R <- Ra         (tbD2<-1, wR<-1)
Ra <- R & Rb    (tbD3<-1, tbA<-1, wD<-1)

>>AND with Base Indexed Addressing       (andx)
=============================================


15____13_____10_______7_____5________2___0__
|0|1| |0|1|0| |d|s|t| |1|0| |s|r|c1| |s|r|c2|
-------------------------------------------

_____________________________________________
|               IMMEDIATE                   |
```

```
-----------------------------------------------

andx Ra, 10( Rb, Rc)

MAB <- PC       (tbP<-1,wMAB<-1)
MDB <- M[MAB]   (wMDB<-1)
R <- MDB        (tbDB<-1, wR<-1)
T <- R + Rb     (tbD2<-1, tbA<-1, wT<-1)
R <- Rc         (tbD3<-1, wR<-1)
MAB <- T + R    (tbT<-1, tbA<-1, wMAB<-1)
MDB <- M[MAB]   (wMDB<-1)
R <- Ra         (tbD2<-1, wR<-1)
Ra <- R & MDB   (tbMDB<-1, tbA<-1, wD<-1)
PC <- PC+1      (tbP<-1, P/M<-1, tbM<-1, wPC<-1)


>>AND with Memory Indirect Addressing   (andn)
===============================================


15____13______10_______7_____5________2____0_
|0|1| |0|1|0| |d|s|t| |1|1| |s|r|c1| |s|r|c2|
-----------------------------------------------
_________________________________________________
|               IMMEDIATE                       |
-------------------------------------------------

andn Ra, @10( Rb, Rc)

MAB <- PC       (tbP<-1,wMAB<-1)
MDB <- M[MAB]   (wMDB<-1)
R <- MDB        (tbDB<-1, wR<-1)
T <- R + Rb     (tbD2<-1, tbA<-1, wT<-1)
R <- Rc         (tbD3<-1, wR<-1)
MAB <- T + R    (tbT<-1, tbA<-1, wMAB<-1)
MDB <- M[MAB]   (wMDB<-1)
MAB <- MDB      (tbMDB<-1, tbB<-1, wMAB<-1)
MDB <- M[MAB]   (wMDB<-1)
R <- Ra         (tbD2<-1, wR<-1)
Ra <- R & MDB   (tbMDB<-1, tbA<-1, wD<-1)
PC <- PC+1      (tbP<-1, P/M<-1, tbM<-1, wPC<-1)


>>OR Immediate                          (ori)
===============================================


15____13______10_______7____5____2_0_
|0|1| |0|1|1| |d|s|t| |0|0| |||| ||||
---------------------------------------
_____________________________________
|              IMMEDIATE              |
-------------------------------------

ori Ra, #5

R <- Ra         (tbD2<-1, wR<-1)
MAB <- PC       (tbP<-1,wMAB<-1)
MDB <- M[MAB]   (wMDB<-1)
Ra <- MDB | R   (tbMDB <- 1, tbA<-1, wD<-1)


>>OR Register                           (orr)
===============================================
```

```
15____13______10_______7_____5______2_0_
|0|1| |0|1|1| |d|s|t| |0|1| |s|r|c| ||||
---------------------------------------

orr Ra, Rb

R <- Ra          (tbD2<-1, wR<-1)
Ra <- R | Rb     (tbD3<-1, tbA<-1, wD<-1)


>>OR with Base Indexed Addressing       (orx)
==============================================


15____13______10_______7_____5________2___0__
|0|1| |0|1|1| |d|s|t| |1|0| |s|r|c1| |s|r|c2|
-----------------------------------------------
_______________________________________________
|               IMMEDIATE                     |
-----------------------------------------------

orx Ra, 10( Rb, Rc)

MAB <- PC        (tbP<-1,wMAB<-1)
MDB <- M[MAB]    (wMDB<-1)
R <- MDB         (tbDB<-1, wR<-1)
T <- R + Rb      (tbD2<-1, tbA<-1, wT<-1)
R <- Rc          (tbD3<-1, wR<-1)
MAB <- T + R     (tbT<-1, tbA<-1, wMAB<-1)
MDB <- M[MAB]    (wMDB<-1)
R <- Ra          (tbD2<-1, wR<-1)
Ra <- R | MDB    (tbMDB<-1, tbA<-1, wD<-1)
PC <- PC+1       (tbP<-1, P/M<-1, tbM<-1, wPC<-1)


>>OR with Memory Indirect Addressing    (orn)
==============================================


15____13______10_______7_____5________2____0_
|0|1| |0|1|1| |d|s|t| |1|1| |s|r|c1| |s|r|c2|
-----------------------------------------------
_______________________________________________
|               IMMEDIATE                     |
-----------------------------------------------

orn Ra, @10( Rb, Rc)

MAB <- PC        (tbP<-1,wMAB<-1)
MDB <- M[MAB]    (wMDB<-1)
R <- MDB         (tbDB<-1, wR<-1)
T <- R + Rb      (tbD2<-1, tbA<-1, wT<-1)
R <- Rc          (tbD3<-1, wR<-1)
MAB <- T + R     (tbT<-1, tbA<-1, wMAB<-1)
MDB <- M[MAB]    (wMDB<-1)
MAB <- MDB       (tbMDB<-1, tbB<-1, wMAB<-1)
MDB <- M[MAB]    (wMDB<-1)
R <- Ra          (tbD2<-1, wR<-1)
Ra <- R | MDB    (tbMDB<-1, tbA<-1, wD<-1)
PC <- PC+1       (tbP<-1, P/M<-1, tbM<-1, wPC<-1)

>>Compare                               (cmp)
==============================================
```

```
15____13_____10___7___5_____2___0_
|0|1| |1|0|0| |||| ||| |||| |r|e|g|
-----------------------------------

reg     : Indicates the register to be compared


***********************************************
* JUMP INSTRUCTIONS             [op-code : 10] *
***********************************************


15____13_____10___7___5____2__0
|1|0| |t|y|p| |||| ||| |||| ||||
--------------------------------

typ     : Specifies the type of jump to be executed


>>Jump Uncoditionally                    (j)
==========================================


15____13_____10___7___5____2__0
|1|0| |0|0|1| |||| ||| |||| ||||
--------------------------------
_________________________________
|         MEMORY ADDRESS        |
---------------------------------

MAB <- PC       (tbP<-1, wMAB<-1)
MDB <- M[MAB]   (wMDB<-1)
PC <- MDB       (tbMDB<-1, wPC<-1)


>>Jump on zero                           (jz)
==========================================

15____13_____10___7___5____2__0
|1|0| |0|1|0| |||| ||| |||| ||||
--------------------------------
_________________________________
|         MEMORY ADDRESS        |
---------------------------------

MAB <- PC                       (tbP<-1, wMAB<-1)
MDB <- M[MAB]                   (wMDB<-1)
if Z is true then PC <- MDB     (tbMDB<-1, wPC<-1)
             else PC <- PC + 1  (tbP<-1, P/M<-1, tbM<-1, wPC<-1)


>>Jump on not zero                       (jnz)
==========================================

15____13_____10___7___5____2__0
|1|0| |0|1|1| |||| ||| |||| ||||
--------------------------------
_________________________________
|         MEMORY ADDRESS        |
---------------------------------

MAB <- PC                       (tbP<-1, wMAB<-1)
MDB <- M[MAB]                   (wMDB<-1)
if Z is false then PC <- MDB    (tbMDB<-1, wPC<-1)
```

```
              else PC <- PC + 1 (tbP<-1, P/M<-1, tbM<-1, wPC<-1)


>>Jump on carry                          (jc)
==========================================

15___13_____10__7__5___2__0
|1|0| |1|0|0| |||| ||| |||| ||||
--------------------------------

_________________________________
|         MEMORY ADDRESS        |
---------------------------------

MAB <- PC                       (tbP<-1, wMAB<-1)
MDB <- M[MAB]                   (wMDB<-1)
if C is true then PC <- MDB     (tbMDB<-1, wPC<-1)
             else PC <- PC + 1  (tbP<-1, P/M<-1, tbM<-1, wPC<-1)


>>Jump on not carry                      (jnc)
==========================================

15___13_____10__7__5___2__0
|1|0| |1|0|1| |||| ||| |||| ||||
--------------------------------

_________________________________
|         MEMORY ADDRESS        |
---------------------------------

MAB <- PC                       (tbP<-1, wMAB<-1)
MDB <- M[MAB]                   (wMDB<-1)
if C is false then PC <- MDB    (tbMDB<-1, wPC<-1)
              else PC <- PC + 1 (tbP<-1, P/M<-1, tbM<-1, wPC<-1)


>>Jump on minus                          (jm)
==========================================

15___13_____10__7__5___2__0
|1|0| |1|1|0| |||| ||| |||| ||||
--------------------------------

_________________________________
|         MEMORY ADDRESS        |
---------------------------------

MAB <- PC                       (tbP<-1, wMAB<-1)
MDB <- M[MAB]                   (wMDB<-1)
if M is true then PC <- MDB     (tbMDB<-1, wPC<-1)
             else PC <- PC + 1  (tbP<-1, P/M<-1, tbM<-1, wPC<-1)


>>Jump on not minus                      (jnm)
==========================================

15___13_____10__7__5___2__0
|1|0| |1|1|1| |||| ||| |||| ||||
--------------------------------

_________________________________
|         MEMORY ADDRESS        |
---------------------------------

MAB <- PC                       (tbP<-1, wMAB<-1)
MDB <- M[MAB]                   (wMDB<-1)
if M is false then PC <- MDB    (tbMDB<-1, wPC<-1)
              else PC <- PC + 1 (tbP<-1, P/M<-1, tbM<-1, wPC<-1)
```

```
*********************************************
* SUBROUTINE CALL AND RETURN    [op-code : 11] *
*********************************************



15____13____10___7___5_____2___0_
|1|1| |t||| |||| ||| |||| |l|n|k|
---------------------------------

t       : Differentiates between a Subroutine call and return
lnk     : The linking register


>>Jump and Link Instruction             (jal)
===========================================


15____13____10___7___5_____2___0_
|1|1| |1||| |||| ||| |||| |l|n|k|
---------------------------------

_________________________________
|         MEMORY ADDRESS        |
---------------------------------

jal Ra, #100

R <- PC + 1     (tbP<-1, P/M'<-1, tbM<-1, wR<-1)
Ra <- R         (wD<-1)
MAB <- PC       (tbP<-1, wMAB<-1)
MDB <- M[MAB]   (wMDB<-1)
T <- PC         (tbP<-1, wT<-1)
R <- T          (tbT<-1, wR<-1)
T <- R + MDB    (tbMDB<-1, tbA<-1, wT<-1)
PC <- T         (tbT<-1, wPC<-1)



>>Return Instruction                    (jr)
===========================================


15____13____10___7___5_____2___0_
|1|1| |1||| |||| ||| |||| |l|n|k|
---------------------------------

jr Rc

MAB <- Rc       (tbD3<-1, tbB<-1, wMAB<-1)
MDB <- M[MAB]   (wMDB<-1)
PC <- MDB       (tbMDB<-1, wPC<-1)
```