


```
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

```
>>
>> Procedure: LookUp_Rom
>>
>> This procedure is used to determine which of five 4k
>> banks of program space the user wishes to execute within.
>>
>> Inputs:
>>       HighAddress : BYTE ( !r0 )
>>
>> Outputs: none
>>
>> Algorithm:
>>
>> BEGIN
>>   CASE HighAddress OF
>>     0 : Adr13 := 0, Adr12 := 0
>>     1 : Adr13 := 0, Adr12 := 0
>>     2 : Adr13 := 0, Adr12 := 1
>>     3 : Adr13 := 1, Adr12 := 0
>>     4 : Adr13 := 1, Adr12 := 1
>>   OTHERWISE Abort
>> END
```

```
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

```
  .LSTOFF
  .FIN
  .DO Internal
  .LSTON
  .Page
  .FIN
```

```
LookUp_Rom:  Tm    !r0,*$80 ;check for adr out of range
             Jr    Nz,LU_Abort
             And   !r0,*$70 ;mask off unnecessary stuff from address
             Cp    !r0,*$40
             Jr    Le,BC_1
```

```
LU_Abort:   Call   Abort
```

```
BC_1:      Ld    !r2,*.HIBYTE. Rom_Table
           Ld    !r3,*.LOWBYTE. Rom_Table
           Swap  !r0 ;turn highaddress into index value
           Rl    !r0 ;multiply index by 2 { 2 byte/element table }
           Add   !r3,!r0 ;index into table
           Adc   !r2,*0
           Ldc   !r1,@!!r2 ;get rom address values
           Incw  !!r2
           Ldc   !r0,@!!r2

           Ld    !r2,*.HIBYTE. RomBank0
           Ld    !r3,!r0
           Lde   @!!r2,!r3 ;set EpromBank0,1
           Ld    !r2,*.HIBYTE. RomBank2
           Ld    !r3,!r1
           Lde   @!!r2,!r3 ;set EpromBank2,3
           Ret
```

```
RomTable:  .DB    0,0 ;Adr13 := 0, Adr12 := 0
           .DB    0,0 ;Adr13 := 0, Adr12 := 0
           .DB    0,1 ;Adr13 := 0, Adr12 := 1
           .DB    1,0 ;Adr13 := 1, Adr12 := 0
           .DB    1,1 ;Adr13 := 1, Adr12 := 1
```

