

- Ime funkcije treba da u sebi sadrži i ime modula na sledeći način:

```
# ac3 decoder function name example
X_S_ac3GetNumberOfChannels
```

Primer zaglavlja funkcije:

```
#####
/**/
*** @Author  Authors first and last name
*** @Date 05. March 2009.
***
*** Function:
*** X_S_MyFunction
***
*** @param list of input and output parametars
***
*** @return where is located return value (if exist)
***
*** Trashes: List of resources trashed inside function
*** Comment: Short description of "What the function is doing"
***
*** E-mail: Authors email
***/
**void X_S_MyFunction()
**{
**
**}
#####
X_S_MyFunction:
#-----
#TODO: Place Function Body Here.
nop
nop
#-----
MyFunction_End:
ret
#####
```

11.3 Imena labela i promenljivih

Labelama uvek treba davati smisljena imena. Treba izbegavati labele koje u sebi imaju na primer jedno slovo, jer ako se takve labele nagomilaju kod praktično postaje nemoguće za održavanje. Primer pogrešne i pravilne labele dat je u tabeli 11.1.

Konvencije za davanje imena promenljivama, funkcijama i labelama :

- Ime promenljive/bafera treba da u sebi sadrži i ime modula na sledeći način:

```
# ac3 decoder buffer name example
X_BX_ac3RightShiftTable
```

- Definisanje promenljive

```
X_VX_skracenicaModulaVariable1 # promenljiva u X memorijskoj zoni
X_VY_skracenicaModulaVariable2 # promenljiva u Y memorijskoj zoni
X_VL_skracenicaModulaVariable2 # promenljiva u L memorijskoj zoni
```

- Definisanje bafera

```
X_BX_skracenicaModulaBuffer1 # bafer definisan u X memorijskoj zoni
X_BY_skracenicaModulaBuffer2 # bafer definisan u Y memorijskoj zoni
X_BL_skracenicaModulaBuffer2 # bafer definisan u L memorijskoj zoni
```

- Definisanje konstante

```
X_CX_skracenicaModulaConst1    # konstanta definisana u X memorijskoj zoni
X_CY_skracenicaModulaConst2    # konstanta definisana u Y memorijskoj zoni
X_CL_skracenicaModulaConst2    # konstanta definisana u L memorijskoj zoni
```

- Definisanje tabela sa konstantnim vrednostima

```
X_TX_skracenicaModulaConst1    # tabela definisana u X memorijskoj zoni
X_TY_skracenicaModulaConst2    # tabela definisana u Y memorijskoj zoni
X_TL_skracenicaModulaConst2    # tabela definisana u L memorijskoj zoni
```

Prilikom davanja imena promenljivama ili baferima, važi pravilo za interne (I_VX_) i eksterne (X_VX_).

Tabela 11.1: primeri labela

Pravilna labela:	Pogrešna labela:
<pre>do(i4),>SampleLoop a0 = xmem[X_VX_LimitLev] # Limit level b0 = ymem[i0]; i0+=1 # cur. Sample a0 - b0 # limit sample? if (a>=0) jmp >DoNotLimitSample # if we are here current sample # is bigger than level, so limit it a0 = b0 %DoNotLimitSample: # ... # additional processing # ... %SampleLoop: nop</pre>	<pre>do(i4), >f a0 = xmem[X_VX_LimitLevel] b0 = ymem[i0]; i0+=1 a0 - b0 if (a>=0) jmp >k a0 = b0 %k %f: nop</pre>

Svi pokazivači (na statičku memoriju ili na heap) moraju biti definisani u X memorijskoj zoni, tako da se na kraj imena promenljive doda sufiks koji označava na koju memorijsku zonu pokazuje:

```
X_VX_tmpTestBuff_Yptr # pointer u X memorijskoj zoni koji pokazuje na
                      # bafer X_BY_tmpTestBuff_deda
                      # alociran u Y memorijskoj zoni

X_VX_tmpTestBuff_Xptr # pointer u X memorijskoj zoni koji pokazuje na
                      # bafer X_BX_tmpTestBuff_baba
                      # alociran u Y memorijskoj zoni
```

Ukoliko je u pitanju pojedinačni pokazivač treba koristiti prefiks X_V?_, dok ako je u pitanju niz pokazivača koristi se prefiks X_B?_.

Za kontrolu toka programa (petlje, uslovni skokovi) poželjno je koristiti lokalne labela (počinju sa znakom „%“). Njihova imena treba da su smisljena ali ne predugačka. Dobar primer bi bilo korišćenje generičkih kombinacija imena (if, %else, %endif) i (do, %loop), pri čemu se ovi simboli mogu eventualno prošiti (npr %loop_ch, %loop_sample i sl.).