GCC, Passing Parameters to Assembly Subroutines

The basic types to pass are 8 bit quantities (bytes, chars) and 16 bit quantities (ints, pointers). All parameters except the first are passed on the stack, starting with an offset of 4 bytes. Function return values are passed in the first pseudo-register (ZD0).

The typical prologue and epilogue of the assembly subroutine is:

```
pshy      ;save the stack frame, could use x
tsy       ;set current stack frame
.         ;your code here
.
.
puly      ;epilogue, restore stack frame
rts       ;return from the function
```

The first parameter is available in register ZD0. If the first parameter is a 16 bit quantity, ZD0 is used. If the first parameter is an 8 bit quantity, ZD0+1 is used. The second through N^{th} parameter is on the stack, starting with an offset of 4 bytes. If a subroutine has five integer parameters, for example, the prologue, epilogue, and access to the parameters would be:

```
pshy
tsy
ldd *ZD0    ;get the first 16 bit parameter
.
.
.
ldd 4,y     ;get the second 16 bit parameter
.
.
.
ldd 6,y     ;get the third 16 bit parameter
.
.
.
ldd 8,y     ;get the fourth 16 bit parameter
.
.
.
ldd 10,y    ;get the fifth 16 bit parameter
.
.
.
puly
rts
```

A function that accepts five 8 bit parameters is similar, with a slightly different offset on the stack:

```
pshy
tsy
ldab *ZD0+1 ;get the first 8 bit parameter, could use A
.
.
.
ldab 4,y    ;get the second 8 bit parameter
.
.
.
ldab 5,y    ;get the third 8 bit parameter
.
.
.
ldab 6,y    ;get the fourth 8 bit parameter
.
.
.
ldab 7,y    ;get the fifth 8 bit parameter
.           ; note:  offsets are 1 byte
.
.
puly
rts
```

Steven L. Barnicki-1998