

PLAYSTATION TECHNICAL NOTE

Date:
Ref:
Author:

Subject: Global Register Allocation

ABSTRACT

It is possible to reserve a register to hold/point to a global variable throughout all the modules of your code. This obviously makes access to the variable very fast but there are a limited number of registers available.

DETAILS

One way to use this on the Playstation is to use a register to point to a structure stored in the high speed data cache memory. Accesses to elements of this structure will then be made by displacements from this register rather than by loading the absolute address of the variable. This is similar to the way that the global pointer register optimisation works except that it gives you control of exactly which variables are placed in this memory.

The register used should be one of the saved registers \$16 - \$23 (s0-s7). If you are not using the GP register optimisation and you specify -G0 when compiling your code then you could use register \$28 (gp) for this purpose.

Here is an example:

```
struct fast_data
{
    int i;
    char z[16];
};

register struct fast_data *f asm("$23");

void main ()
{
    ...

    f->i = 3;

    ...
}
```

Here the assignment `f->i = 3` will code as

```
li    $8,3
sw    $8,0($23)
```

rather than

```
li    $8,3
sw    $8,f
```

where the `sw` instruction expands to

```
lui    $1,(f+$8000)>>16
sw    $8,f&0xffff($1)
```

Points to note:

You can't generate a pointer to something held in a register.

It is up to the programmer to initialise the register. This could be done as follows :

```
void main()
{
    asm("li $23,0x1fff0000");
}
```

or whatever.

The compiler will not make any other use of the register in modules in which the declaration appears. In modules that were compiled without the declaration (e.g. libraries), the compiler may have generated code that makes use of the register. This is not generally a problem since the `s0 - s7` registers must be saved and restored by any function that makes use of them. However, the register may not be set correctly in an event handler or call back or any other function which is called through a pointer by code that was not aware of the global register allocation (e.g. library code). It will therefore be necessary to reload the register if the variables need to be accessed in these types of routines. This is similar to the problem with the global register.

--

support@snsys.com