

HOW TO DO A PROM MICROCODE INTEGRATION

Aug 22, 1980

Filed on: [Idun] <Proms>HowToIntegrateProms.press

Introduction

You can use this procedure to :

- Blow some proms using the **latest** (i.e. most recent) **PromMicrocode release**.
- Blow some proms using a **specified PromMicrocode release**. E.g. the Rev-K release.
- Blow some proms using the **current prerelease PromMicrocode** .
- Do a PromMicrocode Integration. Modify the current prerelease PromMicrocode, blow test proms, and - after testing - save the modified microcode as the current version.

Release versions of Prom files are stored in sub-directories designated by their revision letter. As of Aug 22, 1980 :

revision I is stored on [Idun]<AD0>Proms>Rev-I> and
revision K is stored on [Idun]<AD0>Proms>Rev-K>

Prerelease versions of Prom files are stored on [Idun]<Proms>.

Libjects are used to provide librarian control of prerelease versions of the following PromMicrocode files:

Boot.mc
D0CSProms.bcpl
EtherBoot.mc
Model31Boot.mc
PromDefs.mc
SA4000Boot.mc

CSLProm.cm
TORProm.cm
SDDProm.cm
SPGProm.cm

This procedure is an interim hack until a better way to assign Processor IDs is created.

1. Create a Prom Disk

You will need to create a Prom disk. The best thing to do is to start with a clean disk. Boot the NetExec and type NewOs. Use the long installation dialog and erase the disk. To retrieve the files you will need to create a Prom disk, Always retrieve the following file:

[Idun]<WD0>Do.run

If you want to use **the latest released version** , retrieve:

```
[Idun]<AD0>Proms>Rev-latest>PromDisk.do
```

If you want to use a **specific release** , retrieve:

```
[Idun]<AD0>Proms>Rev-l>PromDisk.do  
or  
[Idun]<AD0>Proms>Rev-K>PromDisk.do
```

If you want to use (or modify) the **current prerelease version** of the PromMicrocode, retrieve:

```
[Idun]<Proms>PromDisk.do
```

Always Type:

```
Do PromDisk
```

This procedure will prompt you for the name of a file server where it can get <Alto> files (e.g., Iris, Isis).

If you only want to blow proms (i.e. you don't want to modify prerelease PromMicrocode), go to step 7.

2. Check Out the Modules You Need to Update

To check out a module to edit, type a command of the form:

```
Access "Reason"/r Module.mc/O
```

where Reason is the reason for checking out the module and Module is the name of your module. This command will retrieve the source.

3. Edit Your Modules

4. Assemble Your Modules

To assemble all modules, type:

```
@MicroAll
```

5. Bind All Modules

To bind all modules, type:

```
@MicroDAI1
```

This creates Boot.mb, which is input to the next step.

6. Make MB File for Pnew

Type:

D0CSProms

This program converts Boot.mb to a form acceptable to Pnew. The output file is named Proms.mb.

7. Check Out the Processor ID

The processor ID is specified by the command line to Pnew. This is an interim hack to keep track of Processor IDs. Currently, Processor IDs are assigned as follows:

CSL	0 - 4,999
TOR	5,000 - 9,999
SDD	10,000 - 19,999
SDD	20,000 - 20,255 (first Dandelion Processor IDs)
SPG	20,256 - ?

You must check out the appropriate command file. Type one of the following:

```
Access "ID"/r CSLProm.cm/O
Access "ID"/r TORProm.cm/O
Access "ID"/r SDDProm.cm/O
Access "ID"/r SPGProm.cm/O
```

This will check out the command file and retrieve it.

8. Blow the C Prom. Use appropriate command file to run Pnew.

The previous steps could have been done on any development system. Step 8 and 9 must be done on a system with an attached Prom Blower. To blow the C Prom (the one that contains the Processor ID), type one of the following:

```
@CSLProm
@TORProm
@SDDProm
@SPGProm
```

When the C Prom has been blown, you will end up in Bravo with this command file in the window. The Processor ID is specified by the number/W parameter. This is a decimal number. Increment this number by one and Put the file.

It is very important that Processor IDs not be duplicated. Please keep the following points in mind when blowing C Proms:

Do not copy or duplicate the C Prom.

Do not type "R" to Pnew to blow another Prom.

It is OK to skip a few Processor ID numbers. If in doubt, skip one.

Don't try to reuse Processor ID numbers when you blow a new revision level. Throw away the old numbers and get new ones.

9. Blow the G Prom. Use command file to run Pnew.

To blow the G Prom, type:

```
@GProm
```

The G Prom can be duplicated. Once you have one, it is easier to duplicate it than to run Pnew each time.

10. Checkin Your Processor ID

You must check in the command file that contains the Processor ID. This file is always left with the number of the next ID to be assigned. Type one of the following commands:

```
Access CSLProm.cm/l  
Access TORProm.cm/l  
Access SDDProm.cm/l  
Access SPGProm.cm/l
```

If you are just blowing some Proms, you are finished !

If you are doing a Prom Integration, go on to steps 11 & 12.

11. Checkin Your Modules

You must check in each module that you have checked out to edit. For each module, type a command of the form:

```
Access Mumble.mc/S
```

where Mumble is the name of your module. This copies the new source back to [Idun]<Proms>.

12. Save Integration Output

If you modified Prom microcode, save the output generated by this integration in [Idun]<Proms> by typing:

```
@Save
```