Cheap-Ass Low-Level DSP Programming Notes

for a suitably hack-up AA3000

by Dave Haynie

The DSP3210 is controlled by two control registers. These registers govern reset of the DSP and interrupt traffic between the DSP and the CPU. The registers are as follows:

Register	Direction	Size
$00dd005c	Read	Longword
$00dd0080	Write	Byte

Both registers have the same bit definitions, at least where a particular bit is defined. This Longword vs. Byte strangeness is due to a bug in the DMAC+'s DSP Control Register; we expect future implementations to look a little more reasonable. The bit assignments are as follows:

Bit	Direction	Function
7	Read/Write	DSP RESET. This bit comes up set low. When low, the DSP is in reset, when high, out of reset.
6	N/A	
5	Read/Write	MASK INT6. This bit comes up set low. When low, INT6* from the DSP to the CPU is masked out, when high, the DSP may cause a CPU level 6 interrupt.
4	Read/Write	MASK INT2. This bit comes up set low. When low, INT2* from the DSP to the CPU is masked out, when high, the DSP may cause a CPU level 2 interrupt.
3	Read Only	CPU INT6. This bit reads low when the DSP is trying to cause a level 6 CPU interrupt, high otherwise.
2	Read Only	CPU INT2. This bit reads low when the DSP is trying to cause a level 2 CPU interrupt, high otherwise.
1	Read/Write	DSP INT1. This bit is written low when the CPU wants to cause a level 1 DSP interrupt. It stays low until acknowledged by the DSP.
0	Read/Write	DSP INT0. This bit is written low when the CPU wants to cause a level 0 DSP interrupt. It stays low until acknowledged by the DSP.