RAM Register Map for buffered DSP values

This table shows the register map of the first page of RAM in the SNES. The values in these registers are used to store important configuration values for the correct function of the DSP. Values that correspond to a specific register in the DSP can be transferred directly, while other values must be collected to a single byte or word and will then be transferred to the DSP. These registers are mirrored on banks $00 to $3F and again on bank $7E (according to SNES memory map). The whole map takes up 144 bytes of memory, while only 54 bytes are being used. Green marks indicate free memory. The grey area is reserved for more audio channels in future updates.

	Description	L CH Vol	R CH Vol	LO 8bit PITCH	HI 8bit PITCH	SOURCE	ADSR1	ADSR2	KEY ON	PITCH MOD	NOISE ENABLE	ECHO ENABLE	tbd	tbd	tbd	tbd	tbd
	DSP register	00	01	02	03	04	05	06									
CH 1	RAM address	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	100A	100B	100C	100D	100E	100F
	value range	0-7F	0-7F	0-FF	0-3F	0-255	FF	E0	1	0	0	0					
	default																

	Description	L CH Vol	R CH Vol	LO 8bit PITCH	HI 8bit PITCH	SOURCE	ADSR1	ADSR2	KEY ON	PITCH MOD	NOISE ENABLE	ECHO ENABLE	tbd	tbd	tbd	tbd	tbd
	DSP register	10	11	12	13	14	15	16									
CH 2	RAM address	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	101A	101B	101C	101D	101E	101F
	value range																
	default																

	Description	L CH Vol	R CH Vol	LO 8bit PITCH	HI 8bit PITCH	SOURCE	ADSR1	ADSR2	KEY ON	PITCH MOD	NOISE ENABLE	ECHO ENABLE	tbd	tbd	tbd	tbd	tbd
	DSP register	20	21	22	23	24	25	26									
CH 3	RAM address	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	102A	102B	102C	102D	102E	102F
	value range																
	default																

	Description	L CH Vol	R CH Vol	LO 8bit PITCH	HI 8bit PITCH	SOURCE	ADSR1	ADSR2	KEY ON	PITCH MOD	NOISE ENABLE	ECHO ENABLE	tbd	tbd	tbd	tbd	tbd
	DSP register	30	31	32	33	34	35	36									
CH 4	RAM address	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039	103A	103B	103C	103D	103E	103F
	value range																
	default																

	X0	X1	X2	X3	X4	X5	X6	4C/5C	2D	3D	4D						
	L CH Vol	R CH Vol	LO 8bit PITCH	HI 8bit PITCH	SOURCE	ADSR1	ADSR2	KEY ON	PITCH MOD	NOISE ENABLE	ECHO ENABLE						
	00	01	02	03	04	05	06										
CH 5	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	100A	100B	100C	100D	100E	100F	

	X0	X1	X2	X3	X4	X5	X6	4C/5C	2D	3D	4D						
	L CH Vol	R CH Vol	LO 8bit PITCH	HI 8bit PITCH	SOURCE	ADSR1	ADSR2	KEY ON	PITCH MOD	NOISE ENABLE	ECHO ENABLE						
	00	01	02	03	04	05	06										
CH 6	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	100A	100B	100C	100D	100E	100F	

	X0	X1	X2	X3	X4	X5	X6	4C/5C	2D	3D	4D						
	L CH Vol	R CH Vol	LO 8bit PITCH	HI 8bit PITCH	SOURCE	ADSR1	ADSR2	KEY ON	PITCH MOD	NOISE ENABLE	ECHO ENABLE						
	00	01	02	03	04	05	06										
CH 7	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	100A	100B	100C	100D	100E	100F	

	X0	X1	X2	X3	X4	X5	X6	4C/5C	2D	3D	4D						
	L CH Vol	R CH Vol	LO 8bit PITCH	HI 8bit PITCH	SOURCE	ADSR1	ADSR2	KEY ON	PITCH MOD	NOISE ENABLE	ECHO ENABLE						
	00	01	02	03	04	05	06										
CH 8	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	100A	100B	100C	100D	100E	100F	

	Description	DSP Flags	Key ON	Key OFF	Offset	Noise	Echo	Main Volume L	Main Volume R	Echo Vol L	Echo Vol R	tbd	tbd	tbd	tbd	tbd	tbd
	DSP register	6C	4C	5C	5D	3D	4D	0C	1C	2C	3C						
MASTER	RAM address	1080	1081	1082	1083	1084	1085	1086	1087	1088	1089	108A	108B	108C	108D	108E	108F
	value range																
	default																

ROM Mapping

This table shows an overview of the whole ROM image. All routines are packed in ROM bank 0 without overlapping and I packets of 1024 bytes each.
Important: The 65816 is actually looking at address FFEA and FFEB for the NMI vector, but the MCU firmware redirects this to 4FEA and 4FEB. This is because then everything can be stored in one single ROM bank.
The area “DSP_Value_Sim” is where the MCU will be storing new values for the DSP and the RAM map actually shows this area in detail.

Bank Number	Source File	Section Name	Origin	Size	Description	CPU Adr. Start	CPU Adr. End	ROM Adr. Start	ROM Adr. En	Code Size	Unused
ROM Bank 0	snes_init.asm	InitializeSNESCode	.org 1024 (forced)	150h	Initialization code for the SNES CPU and peripherals.	00:8000	00:FFFF	00:0000	00:7FFF	1CF0h 7408 bytes	630Fh 25359 bytes
	main.asm	MainCode	.org 2048 (forced)	250h	Main routine						
	video_init.asm	DMAPaletteandVRAM	.org 3072 (forced)	50h	Initialization code for everything regarding video output.						
	dsp_stuff.asm	DSPstuff	.org 4096 (forced)	100h	Initialization and setup code for the APU.						
	(main.asm)	CharacterData	.org 5120 (forced)	800h	Character and Tilemap data						
	dsp_array_values_sim.asm	DSP_Value_Sim	.org 17408 (forced)	50h							
	(main.asm)	VBlank	.org 8192 (forced)	50h							
	(main.asm)	RAM_LOOP	.org 9216 (forced)	100h							
	dsp_ram_routines.asm	DSP_RAM_routines	.org 10240 (forced)	E00h							

Offset in Hex	Offset in Dec	Section Name
0000h	0	
0400h	1024	InitializeSNESCode
0800h	2048	MainCode
0C00h	3072	DMAPaletteandVRAM
1000h	4096	DSPstuff
1400h	5120	CharacterData
1800h	6144	
1C00h	7168	
2000h	8192	VBlank
2400h	9216	RAM_LOOP
2800h	10240	DSP_RAM_routines
2C00h	11264	
3000h	12288	
3400h	13312	
3800h	14336	
3C00h	15360	
4000h	16384	
4400h	17408	DSP_Value_Sim
4800h	18432	
4C00h	19456	
5000h	20480	
5400h	21504	
5800h	22528	
5C00h	23552	
6000h	24576	
6400h	25600	
6800h	26624	
6C00h	27648	
7000h	28672	Vector Table