Nintendo Entertainment Analysis and Development Lempel-Ziv SLI Zip Compression

Overview:

The SLI algorithm is a solution devised by Nintendo's Research and Development branch No. 4 to enhance the LZSS algorithm's capabilities for texture and vertex data used by their software titles. Initially, the encoded data had been strategically structured for execution efficiency on their target platform in the mid-nineties: The Nintendo 64.

Officially, there are two known revisions and three distinct structures of encoded SLI data.

Revisions:

Mario - Dictionary word sizes mirror LZSS' length range of 3-18 bytes.

I.D. - 0x4D494F30 (MIO0) -or- 0x534D53523030 (SMSR00)

Formats - Grouped -or- Tabled

Zelda - The predecessor was upgraded to support a range of 3-273 bytes.

I.D. - 0x59617930 (Yay0) -or- 0x59617A30 (Yaz0)

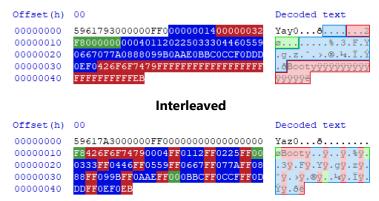
Formats - Tabled -or- Interleaved

Structures:

Grouped

Offset(h)	00	Decoded text
00000000	534D53523030000000000FF0000001E4	SMSR00ðä
00000010	F800F004F013F027F036F04AF05EF06D	ø.ð.ð.ð'ð6ðJð^ðm
00000020	F081F090F0A4F0B80000F0C7F0DBF0EA	ð.ð.ð¤ð,ðÇðÛðê
00000030	F0FEF112F121F135F144F158F16CF17B	ðþñ.ñ!ñ5ñDñXñlñ{
00000040	F18FF19EF1B2F1C6F1D50000F1E9F1F8	ñ.ñžñ°ñÆñÕñéñø
00000050	F20CF220F22FF243F252F266F27AF289	ò.ò ò/òCòRòfòzò‱
00000060	F29DF2ACF2C0F2D4F2E3F2F70000F306	ò.ò⊣òÀòÔòãò÷ó.
00000070	F31AF32EF33DF351F360F374F388F397	ó.ó.ó=óQó`ótó^ó−
08000000	F3ABF3BAF3CEF3E2F3F1F405F4140000	ó«ó°óÎóâóñô.ô
00000090	F428F43CF44BF45FF46EF482F496F4A5	ô(ô<ôKô_ônô,ô−ô¥
0A00000A0	F4B9F4C8F4DCF4F0F4FFF513F522F536	ô¹ôÈôÜôðôÿõ.õ"õ6
000000B0	0000F54AF559F56DF57CF590F5A4F5B3	õJõYõmõ õ.õ¤õ³
000000C0	F5C7F5D6F5EAF5FEF60DF621F630F644	őÇőŐőêőþö.ö!ö0öD
000000D0	F6580000F667F67BF68AF69EF6B2F6C1	öXögö{öŠöžö°öÁ
000000E0	F6D5F6E4F6F8F70CF71BF72FF73EF752	öÕöäöø÷.÷.÷/÷>÷R
000000F0	F766F7750000F789F798F7ACF7C0F7CF	÷f÷u÷‰÷~÷¬÷À÷Ï
00000100	F7E3F7F2F806F81AF829F83DF84CF860	÷ã÷òø.ø.ø)ø=øLø`
00000110	F874F883F8970000F8A6F8BAF8CEF8DD	øtøfø—ø¦ø°øÎøÝ
00000120	F8F1F900F914F928F937F94BF95AF96E	øñù.ù.ù(ù7ùKùZùn
00000130	F982F991F9A5F9B40000F9C8F9DCF9EB	ù,ù'ù¥ù′ùÈùÜùë
00000140	F9FFFA0EFA22FA36FA45FA59FA68FA7C	ùÿú.ú"ú6úEúYúhú
00000150	FA90FA9FFAB3FAC2FAD60000FAEAFAF9	ú.úŸú³úÂúÖúêúù
00000160	FB0DFB1CFB30FB44FB53FB67FB76FB8A	û.û.û0ûDûSûgûvûŠ
00000170	FB9EFBADFBC1FBD0FBE4FBF80000FC07	ûžû.ûÁûĐûäûøü.
00000180	FC1BFC2AFC3EFC52FC61FC75FC84FC98	ü.ü*ü>üRüaüuü"ü~
00000190	FCACFCBBFCCFFCDEFCF2FD06FD150000	ü¬ü»üÏüÞüòý.ý
000001A0	FD29FD38FD4CFD60FD6FFD83FD92FDA6	ý) ý8ýLý`ýoýfý'ý¦
000001B0	FDBAFDC9FDDDFDECFE00FE14FE23FE37	ý°ýÉýÝýìþ.þ.þ#þ7
000001C0	0000FE46FE5AFE6EFE7DFE91FEA0FEB4	pFpZpnp}p'p p'
000001D0	FEC8FED7FEEBFEFAFF0EFF22FF31FF45	pĚþ×þëþúÿ.ÿ"ÿlÿE
000001E0	FF540000FF68FF7CFF8BFF9FFFAEFFC2	ÿTÿhÿ ÿ<ÿŸÿ®ÿÂ
000001F0	FFD64FE5426F6F747900	ÿÖOå <mark>Booty.</mark>

Tabled



Significance:

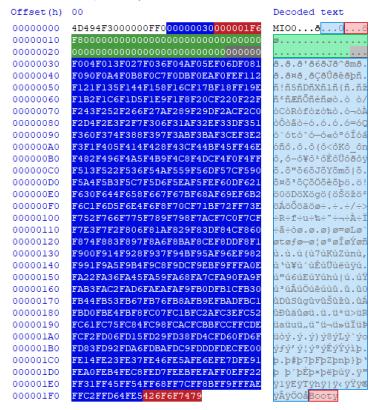
The N64 uses a variant of the MIPS R4300i CPU, a VR4300 from NEC of Japan, which despite its performance capabilities in that era, it will suffer greatly from unaligned load operations (special instructions exist for this situation, but at the cost of performance).

EAD's solution was to structure the encoded data in such a way that load operations interface with sections which are aligned for their respective breadths, so that the decoding process is capable of executing with utmost efficiency as specified in their modus operandi.

The *Grouped* format stores 16-bit HalfWords and Octets in two groups where the HalfWords comprise Bitflags and Dictionaries:



The *Tabled* format stores 32-bit Words, 16-bit HalfWords, and Octets in three delegated partitions of Bitflags, Dictionaries, and Literals; respectively.



At the expense of alignment, the Unused bits effectively serve as padding.

Towards the end of the N64's lifecycle, a formatted octet stream of *Interleaved* coded information had debuted in a few titles before being the final de-facto format that has been in use ever since while the other formats were retired.