

## ELLIOTT 900 SERIES SIMULATOR

### SIR INTERMEDIATE (RELOCATABLE BINARY) FORMAT.

The SIR assembler can produce relocatable binary tapes suitable for loading into ALGOL, 903 FORTRAN or SIR as pre-compiled programs. (Note the relocatable binary format used by 905 FOTRTAN and the 900 Loader is different to that used by ALGOL, 903 SIR and 903 FORTRAN).

The intermediate format is punched to paper tape as seven bit characters.

The loader maintains a dictionary of global symbols encountered and permits forward reference to global labels that have yet to be loaded.

The loader maintains a pointer to the next free location into which data can be loaded, known as the load address.

Whenever a new tape is loaded, the next free location at that point is termed the base address (i.e., the block relative address 0; in SIR) with respect to data on that tape.

In addition to loading words with addresses relative to the base address, the loader also permits loading from an absolute address. This is equivalent to the patch (^) directive in SIR: the relative base address is remembered which loading continues into the locations starting from the patch address. After a restore (loader patch command with address 131071) loading resumes at the previously remembered base address. Patched code can refer to relocated code and vice versa.

An intermediate code tape consists of a series of 21 bit words. Each word consists of three consecutive 7 bit input characters. Bits 21, 20, 19 are interpreted as a command by the loader.

Arbitrary amounts of blank tape (runout) can be interposed before and after commands.

The commands act as follows:

0: PATCH. Remember the current loading address and set an absolute address for loading the next word. (N.B. Not supported by ALGOL loader). Note bit 18 must be a 1 and the address is taken from bits 17-1. If the address is 131071 the loader resumes loading at the previously remembered loading address.

# ELLIOTT 900 SERIES SIMULATOR

21	19	18	1
0	0	0	1
New base address			

1: ABSOLUTE. Store bits 18-1 in the next word. Increment the load address.

21	19	18	1
0	0	1	
Word			

2: RELATIVE. Store bits 18-1 in the next load address and add the base address to the address field of the stored word modulo 8192. Increment the load address.

21	19	18	1
0	1	0	
Word			

3: RESOLVE REFERENCE. (This command is used to fix forward references to local labels, subglobals and literals).

21	19	18	17	16	15	14
0	1	1	0	0	0	A
			Location			
			Offset			
0			0			
			Address			

Bits 14-1 of the command refer to a location to be updated. If bit 15 is 1 the location is treated as an absolute address. If bit 15 is 0, the location is relative to the current base address. Read a further 21 bit word and treat it as an offset (in 2's complement form). Read a third word and treat as the address of the referred to symbol. If the address is zero use the current loading address as the location of the symbol. If the previous contents of the address field in the updated location was other than 8191, indirect through this link to find the next forward reference and update it similarly, working down the chain until terminated by a link address of 8191.

4: GLOBAL. Read a further 21 bit word and treat bits 21-19 of the second word as a sub-command. Treat bits 18-1 of both words as a 6 character global identifier.

# ELLIOTT 900 SERIES SIMULATOR

Sub-command 0 is used to set the loading address relative to a located global symbol (i.e., as required by a SIR directive such as ^LABEL+5, where LABEL is a located global label). The sub-command reads two further words. The first hold the offset, the second is unused.

21	19	18	1
1	0	0	A B C
0	0	0	D E F
offset			
0			

If the sub-command is 1, locate the symbol at the current load address. (If the symbol is already located report an FC error).

Earlier forward references to the global symbol will be resolved by this sub-command.

21	19	18	1
1	0	0	A B C
0	0	1	D E F

If the sub-command is 2 the command is a reference to the specified global symbol. Read two further words. The first word contains modifier and function code bits to be stored at the load address. The second word contains an offset to be added to the address of the global symbol when constructing the address bits to be stored at the current load address. If the global symbol is not yet located, this calculation will be deferred until the symbol is located by a GLOBAL / 2 sub-command later in the input. Increment the load address.

21	19	18	1
1	0	0	A B C
0	1	0	D E F
Modifier + Function			
Offset			

If the sub-command is 3 the global symbol is located at an absolute address. Read two further words. The first word contains the absolute address (as a negative number). The second word is zero. (If the symbol is already located report an FC error. Forward references to the global symbol will be resolved by this sub-command.)

# ELLIOTT 900 SERIES SIMULATOR

21	19	18	1
1	0	0	A B C
0	0	1	D E F
1	1	1	1 Absolute address
0			

An FD error is reported if any other sub-command than 1, 2 or 3 is used.

5: SKIP. Add bits 18-1 to the load address (i.e., skip bits 18-1 locations).

21	19	18	1
1	0	1	Locations

6: END: End of current block. Treat bits 18-1 as a checksum and test against the sum modulo 18 of all characters read since the start of tape or last checksum as appropriate. Error FF is reported if the checksum fails. The next address to be loaded will be treated as the base address (i.e., SIR block relative address 0;) for subsequent loading.

21	19	18	1
1	1	0	Checksum

7: HALT: Stop loading. If bits 18-1 are greater than six, treat them as an address and jump to the specified location. Otherwise set the loader options to equal bits 18-1, prepare the loader to read another tape and pause.

21	19	18	1
1	1	1	Entry Address / Options

Loader options:

- 1: Print store map as code is loaded
- 2: Not known

On encountering HALT, the loader reports unresolved globals as FU errors.

An FA error is reported if the input tape cannot be decoded due to a mispunch or misfeed at any point.

## ELLIOTT 900 SERIES SIMULATOR

An FE error is reported if the store below the loader becomes full.

### Summary of errors

FA	) mispunch or misfeed has corrupted RLB
FD	)
FE	Store full
FF	Checksum failure.