1.1 Branch trace format

x86determiniser writes a branch trace to a file if the environment variable X86D_BRANCH_TRACE contains a file name.

The branch trace is written in text format. Each line contains two 32-bit unsigned integers in hex format, separated by a space. The first integer is a code representing some event within the program. The second integer is the corresponding instruction count since the beginning of the trace.

Within the first integer, the four most significant bits are an “operation code” and the low 28 bits are the “word data”. Each possible 4-bit operation code has a mnemonic name:

Op. Code	**Op. Name**	**Immed. Effect**	**Action**
0	CEM	Yes	Custom event ID
1	BNT	Yes	Not-taken branch (source address = word data | temp)
2	SA	No	Save source address for taken branch (save1 = word data | temp)
3	BT	Yes	Taken branch (target address = word data | value)
4	SM	No	Save middle temp value (temp bits 55 to 28 = word data)
5	SH	No	Save high temp value (temp bits 63 to 36 = word data)
6..12			NOT USED
13..15			RESERVED

When this trace is parsed, the parser must store a temporary variable, which is 64 bits wide. “temp” is a multipurpose variable where bits 27..0 are always 0, and bits 63..28 can be set by SH and SM. These bits are initially 0.

The use of “temp” allows >= 28-bit addresses to be stored within the trace. The trace encoder must generate SH and SM operations as necessary to set the high bits of temp. It’s expected that SH and SM operations will be rare for traces containing only branches, because most branches do not jump over more than 2^28 bytes.

Note that “temp” is persistent – it does not reset to zero after use. To allow random access to a trace, e.g. starting at a specified time, it is important that the values are rewritten to the trace at some regular interval if they are non-zero. This interval is defined as every 10,000 trace elements.