lbi $r0, [$r0+0] (全部為 0 的指令)(ALZ 用途)

0x00000000 在 Andes Core 裡也是一個指令，lbi $r0, [$r0+0]

如果 user 希望在 00000000 發生時通知，(因為可能有異常)可以打開 ALZ 這個 bit。

以下是 ISA manual 裡的說明：

“LBI R0, [R0+0]” baseline version 2 special behavior – This instruction will become a Reserved instruction when the INT_MASK.ALZ (INT_MASK[29]) is set to one. INT_MASK is also named as ir14. This special behavior can be used to debug a system. When this special behavior is used, compiler and assembler should avoid generating this instruction.

下面是 SPA Manual 裡的說明。

9.3.15. Interruption Masking Register

Mnemonic Name: ir14 (INT_MASK)

IM Requirement: Required

Access Mode: Superuser

SR Encoding {Major, Minor, Extension}: {1, 8, 0}

The Interruption Masking Register is used to mask the one software interrupt and hardware interrupts in the Internal Vector Interrupt Controller mode. It also controls if an exception is generated on Divide-by-Zero condition for integer divide instructions.

When MSC_CFG.BASEV <= 1 (baseline version 1 or 2, V1 or V2 architecture)

31	30	29	28 … 17	16	15 … 6	5 … 0
DSSIM	IDIVZE	ALZ	Reserved	SIM	Reserved	H5IM – H0IM (IVIC)

When MSC_CFG.BASEV == 2 (baseline version 3, V3 architecture)

31	30	29	28 … 17	16	15 … 6	5 … 0
DSSIM	IDIVZE	ALZ	Reserved	SIM	H15IM – H6IM (IVIC)	H5IM – H0IM (IVIC)

<table>
<tr>
<td>ALZ</td>
<td>1
(29)</td>
<td>All zero opcode (LBI R0,[R0+0]) reserved instruction exception masking control. This is a baseline version 2 instruction set special behavior and is mainly used for debugging purpose.
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>All zero opcode reserved instruction exception is disabled.</td></tr>
<tr><td>1</td><td>All zero opcode reserved instruction exception is enabled.</td></tr>
</table>
</td>
<td>RW</td>
<td>0</td>
</tr>
</table>

ALZ 是提供一個特殊目的 debug 工具, 設計的用途是如果程式不明原因亂跑時, 如果執行到 All Zero Opcode reserved instruction (0x00000000), 可以盡快產生 exception. 否則如果程式亂跑又 fetch 到的 code 是 All Zero Opcode (一般 padding data 會塞 0x00 or 0xff), 程式會視為正常狀況而繼續執行. 所以 user 可能無法知道程式的錯誤點.

但是, All Zero Opcode 也是一個合法指令 (LBI R0, [R0+0]), 因此在正常的 program 下, ALZ bit 是要 disable. 否則會造成一遇到 lbi $r0, [$r0+0]就發 exception。如果沒有特殊情形，建議把 ALZ disable.