

# Nexus 9000雲端規模ASIC(Tahoe)NX-OS ELAM

## 目錄

[簡介](#)

[適用硬體](#)

[Nexus Tahoe ASIC ELAM程式](#)

[拓撲](#)

[第1步 — 檢驗埠的ASIC、Slice和SrcId](#)

[步驟2 — 連線到模組](#)

[第3步 — 進入ELAM配置模式並從第1步指定適當的ASIC](#)

[第4步 — 配置ELAM觸發器](#)

[第5步 — 使用SRC和DEST IP設定ELAM觸發器](#)

[第6步 — 啟動ELAM](#)

[第7步 — 驗證您的ELAM是否觸發，然後檢視結果](#)

[檢視NX-OS 7.0\(3\)I5\(2\)之前版本的ELAM結果](#)

[VXLAN封裝的ELAM:](#)

[NX-OS 7.0\(3\)I7\(2\)後ELAM配置](#)

[簡介](#)

mELAM(alyzer)Cisco Nexus 9000 CloudScale ASIC

提示：有關ELAM的概述，請參閱[ELAM概述](#)文檔。

[適用硬體](#)

本文檔中涉及的步驟僅適用於以下硬體：

N9K-C93180YC-EX	N9K-C92304QC
N9K-X9736C-EX	N9K-C92300YC
N9K-C93108TC-EX	N9K-X9788TC-FX
N9K-X9732C-EX	N9K-X97284YC-FX
N9K-X97160YC-EX	N9K-C93180YC-FX
N9K-C93180LC-EX	N9K-C93108TC-FX
N9K-C92160YC-X	N9K-C9348GC-

FXP

N9K-C9272Q N9K-X9732C-FX

N9K-C9236C N9K-C9336C-FX2

N9K-C93240YC- N9K-C93300YC-  
FX2 FX2

N9K-C9364C N9K-C9332C

## Nexus Tahoe ASIC ELAM程式

### 拓撲



### 第1步 — 檢驗埠的ASIC、Slice和SrcId

```
N9K-C92160YC-X-2# show hardware internal tah interface e1/49
IfIndex: 436232192
DstIndex: 5952
IfType: 26
Asic: 0 <<<<<<<<< Asic: 0
AsicPort: 56
SrcId: 48 <<<<<<<< Slice: 1 <<<<<<<<< PortOnSlice: 24
```

**注意：**ELAM應只在一個終端視窗上使用，因為您維護每個片段的全域性內容、lu-a2d等。

例如，埠通道(PO)可能有兩個鏈路，Eth 1/53對應於片0，Eth 1/54對應於片1。一次在兩個不同的終端視窗上為不同的片設定ELAM沒有幫助，因為後一個片（例如片1）將覆蓋第一個片（片0），最終在兩個終端視窗上獲得相同的結果。

您可以通過以下方式仔細檢查此資訊：

```
N9K-C92160YC-X-2# show system internal ethpm info interface e1/49 | i i src
IF_STATIC_INFO: port_name=Ethernet1/49,if_index:0x1a006000,ltl=5952,slot=0,
nxos_port=192,dmod=1,dpid=56,
```





**注意事項**中記錄的准則和限制「系統路由模式：template-vxlan-scale」不適用於Cisco NX-OS版本7.0(3)I5(2)及更高版本。將VXLAN BGP EVPN與Cisco NX-OS版本7.0(3)I4(x)或NX-OS版本7.0(3)I5(1)結合使用時，「系統路由模式：以下硬體平台需要template-vxlan-scale」：Cisco Nexus 9300-EX交換器採用X9700-EX線路卡的Cisco Nexus 9500交換器更改「系統路由模式」需要重新載入交換機。

體驗CPU突發流量的範例：

```
module-1(TAH-elam-insel6)# report
```

```
SUGARBOWL ELAM REPORT SUMMARY  
=====
```

```
Incoming Interface: Eth1/3  
Src Idx : 0x9, Src BD : 23  
Outgoing Interface Info: dmod 1, dpid 72  
Dst Idx : 0x601, Dst BD : 802
```

```
Packet Type: IPv4
```

```
Dst MAC address: B0:8B:CF:A3:D0:4B  
Src MAC address: 00:10:DB:FF:10:00  
.1q Tag0 VLAN: 23, cos = 0x0
```

```
Dst IPv4 address: 192.0.2.1  
Src IPv4 address: 192.0.2.2  
Ver      = 4, DSCP      = 2, Don't Fragment = 1  
Proto    = 6, TTL       = 49, More Fragments = 0  
Hdr len  = 20, Pkt len  = 60, Checksum      = 0x63c3
```

```
L4 Protocol : 6  
TCP Dst Port : 80  
TCP Src Port : 46340
```

```
Sup hit: 1, Sup Idx : 2720 <<---- CPU punt, use below CLI to resolve the meaning of Sup Idx
```

```
Drop Info:  
-----
```

```
LUA:  
LUB:  
LUC:  
LUD:  
Final Drops:
```

```
# show system internal access-list sup-redirect-stats all | grep 2720 2720 copp-system-p-acl-  
http 63
```

### **檢視NX-OS 7.0(3)I5(2)之前版本的ELAM結果**

+它是否有dot1q報頭？

```
module-1(TAH-elam-insel6)# report | grep pr_lu_vec_l2v.qtag0 GBL_C++: [MSG] pr_lu_vec_l2v.qtag0_vld:  
0x1 << dot1q yes? 0x1 GBL_C++: [MSG] pr_lu_vec_l2v.qtag0_cos: 0x0 GBL_C++: [MSG]  
pr_lu_vec_l2v.qtag0_de: 0x0 GBL_C++: [MSG] pr_lu_vec_l2v.qtag0_vlan: 0xA << VL 10  
+檢查VLAN:
```

```
module-1(TAH-elam-insel6)# report | grep -1 fpx_lookup_vec.lkup.macsakey.key.fid
```

```
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.vld: 0x1
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.fid_type: 0x0
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.fid_vld: 0x0
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.fid: 0xA << dec 0xa = VL 10
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.mac: 0xFEC80E2715
```

+檢查SRC MAC (實際上在上一步中也可以看到此資訊) :

```
module-1(TAH-elam-insel6)# report | grep -i fpx_lookup_vec.lkup.macsakey.key.mac
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.mac: 0xFEC80E2715 << 00fe.c80e.2715
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.mac: 0xFEC80E2715
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.mac: 0xFEC80E2715
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.mac: 0xFEC80E2715
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.mac: 0xFEC80E2715
```

+這是新學習嗎?

```
module-1(TAH-elam-insel6)# report | grep -i fpx_lookup_vec.sa_notify_info
GBL_C++: [MSG] fpx_lookup_vec.lkup.ptvec.misc1.tcp_flags: 0x0
GBL_C++: [MSG] fpx_lookup_vec.sa_notify_info: 0x5200000C060
GBL_C++: [MSG] fpx_lookup_vec.sa_notify_info.enable: 0x0 << This will be set to 0x1 for learning
to happen
GBL_C++: [MSG] fpx_lookup_vec.sa_notify_info.conv_learn_only: 0x0
```

+檢查SRC和DST IP:

```
module-1(TAH-elam-insel6)# report | grep vec_l3v.ip.*a
GBL_C++: [MSG] pr_lu_vec_l3v.ip.da: 0x0000000000000000c0000201 << DST IP: 192.0.2.1
GBL_C++: [MSG] pr_lu_vec_l3v.ip.sa: 0x0000000000000000c0000202 << SRC IP: 192.0.2.2
```

+驗證輸入SRC\_ID:

```
module-1(TAH-elam-insel6)# report | egrep SRC
GBL_C++: [MSG] SRCID: 0x30
```

```
module-1(TAH-elam-insel6)# report | grep vec.ihdr.ieth.hdr.src_idx
GBL_C++: [MSG] lurw_vec.ihdr.ieth.hdr.src_idx: 0xA9 << sh hardware internal tah int e1/49 | i i niv_idx
```

+如果ELAM未觸發，它將顯示如下內容：

```
module-1(TAH-elam-insel6)# report
GBL_C++: [MSG] tahusd_elam_wrapper_report:27d:asic type 5 inst 0 slice 1 a_to_d 1 insel 6
outsel 0
GBL_C++: [MSG] Inside tahusd_elam_wrapper_dav_report
GBL_C++: [MSG] ELAM not yet triggered <<<<<<
```

### VXLAN封裝的ELAM:

由於會封裝VXLAN封包，因此需要在內部標頭而非外部標頭上觸發ELAM — 請參閱以下有關ARP幀的範例：

```
module-1# debug platform internal tah elam asic 0  
module-1(TAH-elam)# trigger init asic 0 slice 1 in-select 7 out-select 0 use-src-id 48  
module-1(TAH-elam-insel7)# reset  
module-1(TAH-elam-insel7)# set inner arp source-ip-addr 192.0.2.2 target-ip-addr 192.0.2.1  
module-1(TAH-elam-insel7)# start  
module-1(TAH-elam-insel7)# report
```

### **NX-OS 7.0(3)I7(2)後ELAM配置**

在NX-OS 7.0(3)I7(2)之後，現在無需指定ASIC或Slice編號即可全域性觸發ELAM — 請參閱以下示例：

```
Nexus-9K# debug platform internal tah elam  
Nexus-9K(TAH-elam)# trigger init  
Nexus-9K(TAH-elam-insel6)# reset  
Nexus-9K(TAH-elam-insel6)# set outer ipv4 dst_ip 192.0.2.1 src_ip 192.0.2.2  
Nexus-9K(TAH-elam-insel6)# start  
Nexus-9K(TAH-elam-insel6)# report
```

## 關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。