

# Nexus 7000 シリーズ スイッチでのハードウェア転送問題のトラブルシューティング

## 目次

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[トラブルシューティング](#)

[ブレイクアウトケーブルなしで F3 シリーズ モジュールの ELAM を解決して下さい](#)

[ブレイクアウトケーブルで F3 シリーズ モジュールの ELAM を解決して下さい](#)

## 概要

この資料に Cisco Nexus 7000 シリーズ スイッチ用の F3 シリーズ モジュールのハードウェア転送問題を解決する方法を記述されています。

## 前提条件

### 要件

Cisco はこの資料に説明がある情報を続行する前に Cisco Nexus オペレーティング システム (NX-OS) との習熟度がおよび基本的な Nexus アーキテクチャにあることを推奨します。

### 使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づくものです。

- Cisco Nexus 7000 シリーズ スイッチ (N7K)
- Cisco N7K F3 シリーズ モジュール (N7K-F312FQ-25、12 ポート 10/40 ギガビットイーサネットモジュール)
- Cisco NX-OS バージョン 6.2.8a および それ 以降

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。このドキュメントで使用するすべてのデバイスは、クリアな (デフォルト) 設定で作業を開始しています。対象のネットワークが実稼働中である場合には、どのような作業についても、その潜在的な影響について確実に理解しておく必要があります。

## 背景説明

この資料はハードウェアトラブルシューティングのために使用するいくつかの組み込みツールに

フォワーディングテーブルまたはコントロールプレーンのソフトウェア部品を排出したら主に焦点を合わせます。1 そのようなツールは Application-specific integrated circuit (ASIC) の組み込みロジックアナライザモジュール (ELAM) です、シングルパケットをキャプチャし、入力パケットがデータバス (DBUS) および結果 BUS (RBUS) で転送の後でどのように現われるかを示す。

ASIC はフォワーディングパイプラインの中で組み込みであり、パフォーマンスが control plane resources に中断なしでリアルタイムのパケットをキャプチャすることができます。これは質問に答えるのを助けます (以下を参照) :

- パケットはフォワーディングエンジン (FE) に到着しましたか。
- パケットはどんなポートおよび VLAN で受信されますか。
- パケットはどのように現われる (レイヤ2 (L2) またはレイヤ4 (L4) データ) か。
- パケットはどのように変わり、どこで送信されますか。

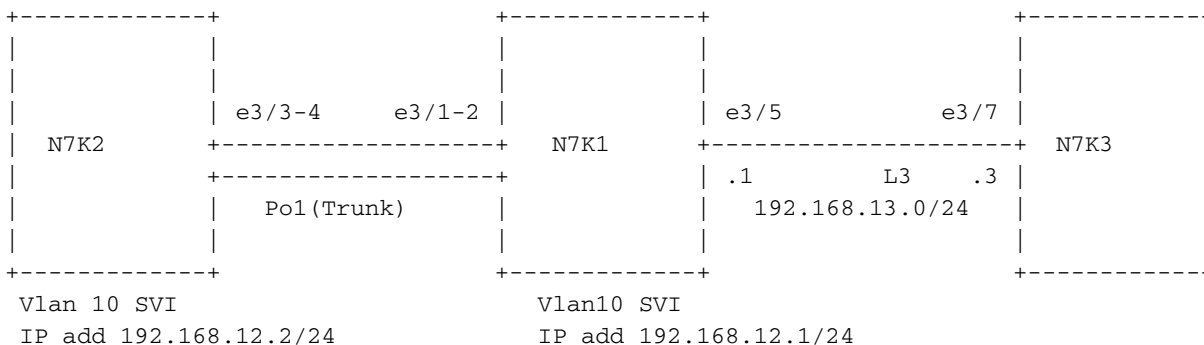
ELAM はハードウェアスイッチングプラットフォームに取り組む Cisco Technical Assistance Center (TAC) エンジニアによって最も広く使われている強力な、粒状の、不侵入ツールです。ただし、ELAM ツールが時間だけに 1 パケットをキャプチャすることを確認することは重要です。すなわち、最初のパケット ELAM が引き起こされた後受け取られる。

## トラブルシューティング

このセクションはブレイクアウトケーブルの使用を含まない、またブレイクアウトケーブルを使用する配置記述します配置の F3 シリーズモジュールの ELAM を解決する方法を。

### ブレイクアウトケーブルなしで F3 シリーズモジュールの ELAM を解決して下さい

これはこのセクション全体の例のために使用するトポロジーです:



このトポロジーについてのいくつかのメモはここにあります:

- N7Ks 実行 NX-OS バージョン 6.2.8a。
- Ping は N7K2 VLAN 10 インターフェイスから 192.168.12.1 のリモートIPアドレスに送られます。
- ELAM は N7K1 のパケットをキャプチャします。

- スロット 3 に挿入される 12 ポート 10/40 ギガビットイーサネットモジュールである N7K-F312FQ-25 は使用されます。

システムのトラブルシューティングを実行し始める前に基本的な接続を確認する必要があります:

```
N7K2# ping 192.168.13.3
PING 192.168.13.3 (192.168.13.3): 56 data bytes
64 bytes from 192.168.13.3: icmp_seq=0 ttl=253 time=1.513 ms
64 bytes from 192.168.13.3: icmp_seq=1 ttl=253 time=1.062 ms
64 bytes from 192.168.13.3: icmp_seq=2 ttl=253 time=0.822 ms
64 bytes from 192.168.13.3: icmp_seq=3 ttl=253 time=0.830 ms
64 bytes from 192.168.13.3: icmp_seq=4 ttl=253 time=0.845 ms

--- 192.168.13.3 ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 0.822/1.014/1.513 ms
```

```
N7K2# show ip route 192.168.13.3
IP Route Table for VRF "default"
'*' denotes best ucast next-hop
***' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>
```

```
192.168.13.0/24, ubest/mbest: 1/0
 *via 192.168.12.1, [1/0], 01:20:36, static
```

!--- The next command verifies the Address Resolution Protocol (ARP) for the next hop.

```
N7K2# show ip arp 192.168.12.1
```

```
----SNIP----
IP ARP Table
Total number of entries: 1
Address      Age      MAC Address  Interface
192.168.12.1 00:10:29 e4c7.2210.a142 Vlan10
```

Supervisor Engine ( Sup ) およびネクスト ホップのためのモジュールで理解しているメディア アクセス制御 ( MAC ) アドレスをまた確認する必要があります:

```
N7K2# show mac address-table address e4c7.2210.a142
```

!--- This command output shows the MAC learning on the Sup (software).

Legend:

\* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC  
age - seconds since last seen,+ - primary entry using vPC Peer-Link,  
(T) - True, (F) - False

VLAN	MAC Address	Type	age	Secure	NTFY	Ports/SWID	SSID	LID
* 10	e4c7.2210.a142	dynamic	120	F	F	Po1		

この出力はモジュール/ハードウェアの MAC ラーニングを示したものです; ただし、インターフェイスを知るために、インデックスを変換して下さい:

```
N7K2# show hardware mac address-table 3 address e4c7.2210.a142
```

FE	Valid	PI	BD	MAC	Index	Stat	SW	Modi	Age	Tmr	GM	Sec
---	-----	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
1	1	1	41	e4c7.2210.a142	0x00a2a	0	0x089	1	185	1	0	0

```
| TR| NT| RM| RMA| Cap| Fld| Always| PV| RD| NN| UC| PI_E8| VIF| SWID| SSWID| LID|
| AP| FY| | | ture| | Learn| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
0 0 0 0 0 0 0 0x00 0 0 1 0 0x000 0x000 0x000 0x00a2a
```

```
N7K2# show system internal pixm info ltl 0x00a2a
```

```
!--- This is the index that was received in the previous output.
```

```
---SNIP---
```

PC_TYPE	PORT	LTL	RES_ID	LTL_FLAG	CB_FLAG	MEMB_CNT
Normal	Po1	0x0a2a	0x16000000	0x00000000	0x00000002	2

```
Member rbh rbh_cnt
Eth3/4 0x000000f0 0x04
Eth3/3 0x0000000f 0x04
```

```
---SNIP---
```

仮想デバイス コンテキスト ( VDC ) 数を得るためにこれらのコマンドを入力して下さい ( この例で、3 ) あり、モジュールの MAC を直接チェックします:

```
N7K2# show vdc
```

```
---SNIP---
```

vdc_id	vdc_name	state	mac	type	lc
3	N7K2	active	e4:c7:22:10:a1:43	Ethernet	f3

```
module-3# attach module 3
```

```
module-3# vdc 3
```

```
!--- This data is obtained from the previous command output.
```

```
module-3# show mac address-table address e4c7.2210.a142
```

```
Legend:
```

```
* - primary entry, G - Gateway MAC, (R) - Routed MAC, (d) - dec
Age - seconds since last seen,,+ - primary entry using vPC Peer-Link
(T) - True, (F) - False, h - hex, d - decimal
```

```
VDC = 3
```

FE	VLAN/BD	MAC Address	Type	Age	Secure	NTFY	Ports/SWID.SSID.LID(d)
* 1	10	e4c7.2210.a142	dynamic	360	F	F	Po1

N7K2 からの Sup のトラフィックを転送するために使用する、またリンクを判別して下さいポートチャンネル 1 のリンクをポートチャンネル 1 が N7K1 から N7K2 への使用されるとき N7K3 からの応答を返すために使用される:

```
N7K2# show port-channel load-balance forwarding-path interface port-channel 1 src-ip 192.168.12.2 dst-ip 192.168.13.3 module 3
```

```
Module 3: Missing params will be substituted by 0's.
```

```
Load-balance Algorithm: src-dst ip
```

```
RBH: 0xd2 Outgoing port id: Ethernet3/3
```

```
N7K1# show port-channel load-balance forwarding-path interface port-channel 1 src-ip
```

### 192.168.13.3 dst-ip 192.168.12.2 module 3

Module 3: Missing params will be substituted by 0's.

Load-balance Algorithm: src-dst ip

RBH: 0xd2      Outgoing port id: Ethernet3/1

N7K2 ( IP アドレス 192.168.12.2 ) からの PING を送信し、パケットが N7K3 ( 192.168.13.3 ) IP アドレスに転送されることを確認するために入方向の N7K1 のパケットをキャプチャして下さい。

PING を送信 する前に、ハードウェア集結のナレッジがあるはずです。集結を理解するためにこれらのステップを完了して下さい:

#### 1. モジュールを接続して下さい:

```
N7K1# attach module 3
Attaching to module 3 ...
To exit type 'exit', to abort type '$.'
```

#### 2. フランカー例を識別して下さい。フランカーは半導体素子 ( F3 シリーズ モジュールのための SOC ) のスイッチ ASIC です。各フランカーはモジュールの 2 つの外部ポートにマッピングされます ( モジュールタイプごとの情報の変更は N7K-F312FQ-25 に特定であり、 )。

モジュールに 12 のポート、および各 ASIC マップがフロントパネルの 2 つのポートへあります、つまりモジュールで利用可能な 6 つの ( 0-5 ) フランカー例があることを意味します ( 例数はゼロベースです )。注: 始める前にネットワーク 管理権限があることを確認して下さい。N7K1 のポート チャンネル 1 で N7K2 から着くパケットをキャプチャするので、各例にマッピングされるポートを探して下さい ( e3/1 および e3/2 ):

```
module-3# show hardware internal dev-port-map
-----
CARD_TYPE:          12 port 40G
>Front Panel ports:12
-----
Device name          Dev role              Abbr num_inst:
-----
>Flanker Eth Mac Driver  DEV_ETHERNET_MAC      MAC_0  6
>Flanker Fwd Driver      DEV_LAYER_2_LOOKUP    L2LKP  6

!--- Check for the L2LKP number for ports 1 and 2.

>Flanker Xbar Driver     DEV_XBAR_INTF         XBAR_INTF 6
>Flanker Queue Driver    DEV_QUEUEING          QUEUE  6
>Sacramento Xbar ASIC    DEV_SWITCH_FABRIC     SWICHF  1
>Flanker L3 Driver       DEV_LAYER_3_LOOKUP    L3LKP  6
>EDC                     DEV_PHY               PHYS  2
+-----+
+-----+++FRONT PANEL PORT TO ASIC INSTANCE MAP+++-----+
+-----+
FP port |  PHYS |  MAC_0 |  L2LKP |  L3LKP |  QUEUE |  SWICHF
-----+-----+-----+-----+-----+-----+
1       |      0 |      0 |      0 |      0 |      0 |      0
2       |      0 |      0 |      0 |      0 |      0 |      0
3       |      1 |      1 |      1 |      1 |      1 |      0
4       |      1 |      1 |      1 |      1 |      1 |      0
5       |      0 |      2 |      2 |      2 |      2 |      0
6       |      0 |      2 |      2 |      2 |      2 |      0
7       |      1 |      3 |      3 |      3 |      3 |      0
8       |      1 |      3 |      3 |      3 |      3 |      0

!--- The L2LKP for both ports is 0, so both belong to instance 0.
```

9	4	4	4	4	0
10	4	4	4	4	0
11	5	5	5	5	0
12	5	5	5	5	0

```

+-----+
+-----+

```

3. 例を選択し、トリガーを設定し、キャプチャを始めて下さい。しかし ELAM トリガーによって使用できる多くのオプションがあること理解することは重要です:

```

module-3# elam asic flanker instance 0
module-3(fln-elam)# layer2
module-3(fln-l2-elam)# trigger ?
  dbus  Pre L2 BUS
  rbus  Post L2 BUS

```

-----SNIP----- これら二つのオプションはスイッチによって受け取られる ) キャプチャ ( に DBUS をパケットたいと思う場合重要です含め。ルックアップに服従しないこれは未加工パケットです。RBUS は DBUS のためのハードウェアでルックアップ結果を示します。完全な ELAM および分析に関しては、RBUS および DBUS を両方キャプチャして下さい。

次の出力は DBUS オプションとキャプチャ することができるパケットの種類を示したものです。この例では、インターネット プロトコル バージョン 4 ( IPv4 ) パケットは選択されず:

```

module-3(fln-l2-elam)# trigger dbus ?
  arp      ARP Frame Format
  fc       Fc hdr Frame Format
  ipv4     IPV4 Frame Format
  ipv6     IPV6 Frame Format
  mpls     MPLS
  other    L2 hdr Frame Format
  pup      PUP Frame Format
  rarp     RARP Frame Format
  valid    On valid packet

```

使用するために選択できるいくつかの追加オプションはここにありません:

```

module-3(fln-l2-elam)# trigger dbus ipv4 ?
  egress          Egress packets

```

!--- Capture packets in egress (outbound from the port).

```

  if              If Trigger Condition
  ingress         Ingress packets

```

!--- Capture packets in ingress (inbound to the port).

```

  multicast          Multicast packet

```

multicast-replication Multicast replication この例ではキャプチャのための条件を選択するためにハンドルが使用される場合。次の出力で示されていて L2、L3 および L4 ヘッダに基づかせているオプションのほとんど。送信元および宛先 IP アドレスもキャプチャのために使用されます。

```

module-3(fln-l2-elam)# trigger dbus ipv4 ingress if ?
<CR>
  acos              Acos
  block-capture     Capture 12 blocks
  bpdu              Bpdu

```

bundle-port	Bundle-port
ccc	Ccc
copp	Copp
da-type	Da-type
de-cfi	De cfi
destination-index	Destination-index
destination-ipv4-address	destination ipv4 address
destination-mac-address	Destination-mac-address
destination-vif	Destination-vif
df	df
dfst	Dfst
dft	Dft
disable-index-learn	Disable-index-learn
disable-new-learn	Disable-new-learn
dont-forward	Dont-forward
dont-learn	Dont-learn
dtag-ftag	Dtag-ftag
dtag-ttl	Dtag-ttl
dti-type-vpnid	Dti type vpnid
error	Error
erspan-kpa-valid	Erspan kpa valid
ff	Ff
frag	frag
header-type	Header type
ib-length-bundle	Ib length bundle
ids-check-fail	Ids-check-fail
ignore-acli	Ignore-acli
ignore-aclo	Ignore-aclo
ignore-qosi	Ignore-qosi
ignore-qoso	Ignore-qoso
inband-flow-creation-deletion	Inband-flow-creation-deletion
index-direct	Index-direct
inner-cos	Inner-cos
inner-de-valid	Inner de valid
inner-drop-eligibility	Inner-drop-eligibility
ip-da-multicast	Ip-da-multicast
ip-multicast	Ip-multicast
ip-multicast-control	Ip-multicast-control
ipv6	Ipv6
l2	L2
l2-frame-type	L2-frame-type
l2-length-check	L2 length check
l2lu-mode	L2lu-mode
l3-packet-length	l3 packet length
l4-protocol	l4 protocol
label-count	Label count
last-ethertype	Last-ethertype
lbl0-eos	Lbl0 eos
lbl0-exp	Lbl0 exp
lbl0-lbl	Lbl0 lbl
lbl0-ttl	Lbl0 ttl
lbl0-valid	Lbl0 valid
lbl1-exp	Lbl1 exp
lbl1-ttl	Lbl1 ttl
mac-in-mac-valid	Mac-in-mac-valid
mc	Mc
md-acos	Md acos
md-destination-table-index	Md destination table index
md-fwd-only	Md fwd only
md-lif	Md lif
md-mark-enable	Md mark enable
md-multicast-bridge-disable	Md multicast bridge disable
md-preserve-acos	Md preserve acos
md-qos-group-id	Md qos group id

md-replication-packet	Md replication packet
md-router-mac	Md router mac
md-ttl-err	Md-ttl-err
md-version	Md version
mf	mf
mim-destination-mac-address	Mim-destination-mac-address
mim-source-mac-address	Mim-source-mac-address
mlh-type	Mlh-type
no-stats	No-stats
notify-index-learn	Notify-index-learn
notify-new-learn	Notify-new-learn
null-label-exp	Null label exp
null-label-ttl	Null label ttl
null-label-valid	Null label valid
option	option
outer-cos	Outer-cos
outer-drop-eligibility	Outer-drop-eligibility
ovl-mlh-bndl	Ovl mlh bndl
ovl-ulh-bndl	Ovl ulh bndl
ovl-ulh-bndl-1	Ovl-ulh-bndl-1
ovl-ulh-bndl-2	Ovl-ulh-bndl-2
packet-length	Packet-length
packet-type	Packet type
pd-tag-gt-2	Pdt-tag-gt-2
pd-tag0	Pdt-tag0
pd-tag1	Pdt-tag1
pd-valid	Pdt-valid
pd-value	Pdt-value
port-id	Port-id
rbh	Rbh
rdt	Rdt
recirc-shim-vxlan-src-peer-id	Recirc shim vxlan src peer id
recirc-acos	Recirc acos
recirc-bypass-ife	Recirc bypass ife
recirc-bypass-l2	Recirc bypass l2
recirc-destination-table-index	Recirc destination table index
recirc-forward-only	Recirc forward only
recirc-l2-tunnel-encap	Recirc l2 tunnel encap
recirc-lif	Recirc lif
recirc-ls-hash	Recirc ls hash
recirc-mark-enable	Recirc mark enable
recirc-multicast-bridge-disable	Recirc multicast bridge disable
recirc-preserve-acos	Recirc preserve acos
recirc-preserve-ls-hash	Recirc preserve ls hash
recirc-preserve-rbh	Recirc preserve rbh
recirc-qos-group-id	Recirc qos group id
recirc-replication-packet	Recirc replication packet
recirc-router-mac	Recirc router mac
recirc-ttl-err	Recirc ttl err
recirc-valid	Recirc-valid
recirc-version	Recirc version
redirect	Redirect
repl-bypass-ife	Repl bypass ife
repl-bypass-l2	Repl bypass l2
repl-disable-local-bridge	Repl disable local bridge
repl-fwd-only	Repl fwd only
repl-l2-tunnel-encap	Repl l2 tunnel encap
repl-l2-tunnel-info	Repl l2 tunnel info
repl-lif	Repl lif
repl-mark-enable	Repl mark enable
repl-met-lif	Repl met lif
repl-m13	Repl m13
repl-preserve-acos	Repl preserve acos
repl-preserve-rbh	Repl preserve rbh



repl-qos-group-id	Repl qos group id
repl-replication-packet	Repl replication packet
repl-router-mac	Repl router mac
repl-ttl-err	Repl ttl err
repl-version	Repl version
rf	Rf
second-inner-cos	Second inner cos
segment-id	Segment id
segment-id-valid	Segment id valid
sequence-number	Sequence-number
sg-tag	Sg-tag
shim-valid	Shim valid
source-index	Source-index
source-ipv4-address	source ipv4 address
source-mac-address	Source-mac-address
source-vif	Source-vif
status-ce-1q	Status-ce-1q
status-is-1q	Status-is-1q
sup-eid	Sup-eid
tos	tos
traceroute	Traceroute
trig	Any of previous elam triggered
trill-encap	Trill-encap
ttl	ttl
tunnel-bundle	Tunnel bundle
tunnel-type	Tunnel type
ulh-type	Ulh-type
valid	VALID
v1	V1
vlan	Vlan
vn-p	Vn p
vn-valid	Vn-valid
vqi	Vqi
vqi-valid	Vqi-valid
vsl-num	Vsl-num

この出力は最終的なトリガー オプションを示した  
 ものです:

```

module-3# elam asic flanker instance 0
module-3(fln-elam)# layer2
module-3(fln-l2-elam)# trigger dbus ipv4 ingress if source-ipv4-address 192.168.12.2
destination-ipv4-address 192.168.13.3

```

```

module-3(fln-l2-elam)# trigger rbus ingress if trig

```

注: RBUS 設定は通常複雑、簡素化される。

4. トリガーをチェックし、**status** コマンドを入力し、キャプチャ プロセスを開始し、PING を N7K2 から N7K3 ( 192.168.12.1 への 192.168.13.3 ) に始めるため:

```

module-3(fln-l2-elam)# stat
ELAM Slot 3 instance 0: L2 DBUS Configuration: trigger dbus ipv4 ingress if
source-ipv4-address 192.168.12.2 destination-ipv4-address 192.168.13.3
L2 DBUS: Configured
ELAM Slot 3 instance 0: L2 RBUS Configuration: trigger rbus ingress if trig
L2 RBUS: Configured

```

```

module-3(fln-l2-elam)# start
module-3(fln-l2-elam)# status

```

```
!--- The status shows as Armed because the process has begun.
```

```

ELAM Slot 3 instance 0: L2 DBUS Configuration: trigger dbus ipv4 ingress if
source-ipv4-address 192.168.12.2 destination-ipv4-address 192.168.13.1

```

```
L2 DBUS: Armed
ELAM Slot 3 instance 0: L2 RBUS Configuration: trigger rbus ingress if trig
L2 RBUS: Armed
module-3(fln-l2-elam)#
```

```
module-3(fln-l2-elam)# status
```

```
!--- If the packet is captured, the status shows Triggered.
```

```
ELAM Slot 3 instance 0: L2 DBUS Configuration: trigger dbus ipv4 ingress if
source-ipv4-address 192.168.12.2 destination-ipv4-address 192.168.13.3
L2 DBUS: Triggered
ELAM Slot 3 instance 0: L2 RBUS Configuration: trigger rbus ingress if trig
L2 RBUS: Triggered
module-3(fln-l2-elam)#
```

5. ステータスが引き起こされて示す場合、RBUS におよび DBUS に両方同じシーケンス番号が同じパケットのためであることを確認するためにあるかどうか確かめて下さい。この例では、0x55 は使用されますが、シーケンス番号を示すカラムは異なっています:

```
module-3(fln-l2-elam)# show dbus | in seq
sequence-number      : 0x6b                v1      : 0x0
```

```
!--- The sequence number is the same (0x6b).
```

```
module-3(fln-l2-elam)# show rbus | in seq
l2-rbus-trigger      : 0x1                sequence-number      : 0x6b
```

6. 提示 D・パスを入力し、DBUS および RBUS を確認するために rbus コマンドを示して下さい。DBUS コマンド 出力の調査資料索引および RBUS コマンド 出力の宛先インデックスを探して下さい:

```
module-3(fln-l2-elam)# show dbus
cp = 0x1007db4c, buf = 0x1007db4c, end = 0x10089e9c
```

```
-----
Flanker Instance 00 - Capture Buffer On L2 DBUS:
```

```
Status(0x0102), TriggerWord(0x000), SampleStored(0x005),CaptureBufferPointer(0x005)
```

```
is_l2_egress: 0x0000, data_size: 0x023
```

```
[000]: 5902a000 08010000 00000000 0cc01400 00145800 00000000 01800100 00000000
00000000 00000000 003931c8 842850b9 31c88428 50c00000 01ac0000 00000000 00000000
00000000 00000000 00000000 00000005 80005000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00605406 01605406 8180008f f0054608 00000000
```

```
Printing packet 0
```

```
-----
L2 DBUS PRS MLH IPV4
-----
```

```
label-count      : 0x0                mc                : 0x0
null-label-valid : 0x0                null-label-exp    : 0x0
null-label-ttl   : 0x0                lb10-vld         : 0x0
lb10-eos         : 0x0                lb10-lb1        : 0x0
lb10-exp         : 0x0                lb10-ttl        : 0x0
lb11-exp         : 0x0                lb11-ttl        : 0x0
ipv4             : 0x0                ipv6             : 0x0
l4-protocol      : 0x1                df               : 0x0
mf               : 0x0                frag             : 0x0
ttl              : 0xff               l3-packet-length : 0x54
option           : 0x0                tos              : 0x0
sup-eid          : 0x0                header-type      : 0x1
error            : 0x0                redirect        : 0x0
```

```

port-id          : 0x0          last-ethertype   : 0x800
l2-frame-type    : 0x0          da-type          : 0x0
packet-type      : 0x0          l2-length-check  : 0x0
ip-da-multicast  : 0x0          ip-multicast     : 0x0
ip-multicast-control: 0x0       ids-check-fail   : 0x0
traceroute       : 0x0          outer-cos        : 0x0
inner-cos        : 0x0          vqi-valid       : 0x0
vqi              : 0x0          packet-length    : 0x66
vlan             : 0xa          destination-index: 0x0
source-index    : 0xa2c       bundle-port    : 0x0
acos            : 0x0          outer-drop-eligibility: 0x0
inner-drop-eligibility: 0x0     sg-tag          : 0x0
rbh             : 0x0          vsl-num         : 0x0
inband-flow-creation-deletion: 0x0 ignore-qoso      : 0x0
ignore-qosi     : 0x0          ignore-aclo     : 0x0
ignore-acli     : 0x0          index-direct    : 0x0
no-stats        : 0x0          dont-forward    : 0x0
notify-index-learn : 0x1       notify-new-learn : 0x1
disable-new-learn : 0x0       disable-index-learn : 0x0
dont-learn      : 0x0          bpdu            : 0x0
ff              : 0x0          rf              : 0x0
ccc             : 0x0          l2              : 0x0
rdt             : 0x0          dft             : 0x0
dfst           : 0x0          status-ce-1q    : 0x0
status-is-1q   : 0x1       trill-encap     : 0x0
mim-valid      : 0x0          dtag-ttl        : 0x0
dtag-ftag      : 0x0          valid           : 0x1
erspan-kpa-valid : 0x0       recir-shim-vxlan-src-peer-id: 0x0
vn-valid       : 0x0          source-vif      : 0x0
destination-vif : 0x0          vn-p            : 0x0
sequence-number : 0x6b       vl              : 0x0
inner-de-valid  : 0x0          de-cfi         : 0x0
second-inner-cos : 0x0       tunnel-type     : 0x0
shim-valid     : 0x0
segment-id-valid : 0x0       copp           : 0x0
dti-type-vpnid : 0x0          segment-id     : 0x0
ib-length-bundle : 0x58000    mlh-type       : 0x5
ulh-type       : 0x6
source-ipv4-address: 192.168.12.2
destination-ipv4-address: 192.168.13.3
mim-destination-mac-address : 0000.0000.0000
mim-source-mac-address : 0000.0000.0000
destination-mac-address : e4c7.2210.a142
source-mac-address : e4c7.2210.a143

```

```

module-3(fln-l2-elam)# show rbus
cp = 0x100a2548, buf = 0x100a2548, end = 0x100ae898

```

```

-----
Flanker Instance 00 - Capture Buffer On L2 RBUS:

```

```

Status(0x0102), TriggerWord(0x000), SampleStored(0x005), CaptureBufferPointer(0x005)

```

```

is_l2_egress: 0x0000, data_size: 0x018
[000]: 0059d930 0000000c c0000000 03580000 00000000 00000000 0000001f 57b00021
fdfc0000 00000000 02000000 14001402 8b000105 00000000 68200000 00000000 00000000
00000400 00008000 005b0000 00fe0e4c 7220850a 210000a0 000000b6

```

```

Printing packet 0

```

```

-----
L2 RBUS INGRESS CONTENT
-----

```

```

pad          : 0x16764          valid          : 0x1

```

```

l2-rbus-trigger      : 0x1          sequence-number      : 0x6b
rit-ipv4-id         : 0x0          ipv4-tunnel-encap   : 0x0
rit-mpls-rw        : 0x0          ml2-ptr             : 0x0
ml3-ptr            : 0x0          mark                : 0x0
result-cap3        : 0x0          di1-v5-delta-length: 0x0
di1-v5-delta-length-plus: 0x0      di1-v4-delta-length : 0x0
di1-v4-delta-length-plus: 0x0      di2-delta-length    : 0x0
di2-delta-length-plus: 0x0          ml2-delta-length    : 0x0
ml2-delta-length-plus: 0x0          ml3-delta-length    : 0x0
ml3-delta-length-plus: 0x0          s-vector            : 0x0
lcpu-ff-valid     : 0x0          sup-di-vqi          : 0x0
erspan-term-index-dir: 0x0        erspan-buffer-check : 0x0
l2-tunnel-decapped : 0x0          13-delta-length     : 0x0
rit-crc16-valid   : 0x1          rit-crc16           : 0xf57b
vntag-p           : 0x0          frr-recirc          : 0x0
ingress-lif       : 0x1          earl-proxy-vld      : 0x0
md-di-vld         : 0x0          rc                  : 0x0
segment-id-valid  : 0x0          ttl-out             : 0xfe
ttl-mid           : 0xfe          tos-out             : 0x0
tos-in            : 0x0          orig-vlan1          : 0x0
vlan1             : 0x0          source-peer-id      : 0x0
final-ignore-qoso : 0x0          port-id             : 0x0
cr-type           : 0x1          pup-packet          : 0x0
bpdu              : 0x0          vdc                 : 0x0
traceroute        : 0x0          de                  : 0x0
cos               : 0x0          inner-drop-eligibility: 0x0
inner-cos         : 0x0          acos                : 0x0
di-ltl-index      : 0x50        13-multicast-di   : 0x50
source-index      : 0xa2c        vlan                 : 0x0
index-direct     : 0x0          di1-valid           : 0x1
vqi              : 0x50          di2-valid           : 0x0
v5-fpoe-idx     : 0x0          di2-fpoe-idx        : 0x0
l3-multicast-v5  : 0x0          dft                 : 0x0
dfst             : 0x0          l3-learning-ff      : 0x0
result-rbh       : 0xd0          di2-cr-type         : 0x0
result-2         : 0x1          dtag-ftag           : 0x0
dtag-ttl         : 0x20          mac-in-mac-op       : 0x0
dvif             : 0x0          result-cap1         : 0x0
result-cap2      : 0x0          erspan-term         : 0x0
erspan-decap     : 0x0          dont-learn          : 0x0
routed-frame     : 0x1          copy-cause          : 0x0
l2-copy-cause    : 0x0          l3-rit-ptr          : 0x5b
sg-tag           : 0x0          trill-nh-id         : 0x0
ttl-in           : 0xfe          fc-up               : 0x0
up-did           : 0x0          did                 : 0xe4c722
up-sid           : 0x0          sid                 : 0x10a144
shim-l2-tunnel-encap: 0x0        shim-ls-hash        : 0x8
shim-rc          : 0x0          shim-lif            : 0x1
shim-replication-pkt: 0x0        shim-router-mac     : 0x1
shim-mark-enable : 0x0          shim-qos-group-id   : 0x0
shim-destination-table-index: 0x5b  shim-acos-preserve  : 0x0
mim-destination-mac-address : 0000.0000.0000
mim-source-mac-address : 0000.0000.0000

```

## 7. Sup の宛先インデックスおよび調査資料索引をチェックして下さい:

```
N7K1# show system internal pixm info ltl 0xa2c
```

PC_TYPE	PORT	LTL	RES_ID	LTL_FLAG	CB_FLAG	MEMB_CNT
Normal	Po1	0x0a2c	0x16000000	0x00000000	0x00000002	2

```

Member rbh rbh_cnt
Eth3/2 0x000000f0 0x04

```

Eth3/1 0x0000000f 0x04

CBL Check States: Ingress: Enabled; Egress: Enabled

VLAN	BD	BD-St	CBL St & Direction:
1	0x15	INCLUDE_IF_IN_BD	FORWARDING (Both)
10	0x19	INCLUDE_IF_IN_BD	FORWARDING (Both)

Member info

Type	LTL
PORT_CHANNEL	Po1
FLOOD_W_FPOE	0x8019
FLOOD_W_FPOE	0x8015

N7K1# **show system internal pixm info ltl 0x50**  
 0x0050 is in DCE/FC pool

Member info

Type	LTL
PHY_PORT	Eth3/5

この出力はパケットがポート チャネル 1 ( Po1 ) で受信され、Eth3/5  
 によって転送されたことを確認します。

### 8. 正しいプログラミングのためのモジュールの Local Target Logic ( LTL ) を確認して下さい:

```
module-3# show system internal pixmc info ltl-cb ltl 0xa2c
ltl |ltl_type|if_index|lc_type| vdc |v4_fpoelv5_fpoel base_fpoel_idx | flag
0x0a2c | 4 | Po1 | 2 | 2 | 0x00 | 0x00 | 0x0000 | 0x0
, local ports:
VDCs the entry is part of:
```

LTL HW programming info

Index	ec	drop	span_vec	SOM	ucr_fab
[ a2c ]	1	0	0	0	0
RBH		VQI		PS (INST:LPOE)	
0,	40		0	:	1
1,	40		0	:	1
2,	40		0	:	1
3,	40		0	:	1
4,	44		0	:	10
5,	44		0	:	10
6,	44		0	:	10
7,	44		0	:	10
8,	0		0	:	1
9,	0		0	:	1
a,	0		0	:	1
b,	0		0	:	1
c,	0		0	:	10
d,	0		0	:	10
e,	0		0	:	10
f,	0		0	:	10

```

module-3# show system internal pixmc info ltl-cb ltl 0x50
ltl |ltl_type|if_index|lc_type| vdc |v4_fpoe | v5_fpoe| base_fpoe_idx | flag
0x0050 | 5 |Eth3/5 | 2 | 2 | 0x00 | 0x00 | 0x0000 | 0x0
, local ports:
VDCs the entry is part of:

```

```

LTL HW programming info
.....
-----
|Index | ec |drop|span_vec|SOM|ucr_fab|
|-----|
|[ 50]| 1| 0| 0| 0| 0|
| RBH | VQI | PS
|-----|
ALL RBH| 50 | 2 : 1

```

9. 出力に ELAM パケットをキャプチャして下さい。パケットをキャプチャするために、IP address 192.168.13.3 から 192.168.12.2 に PING 応答を送って下さい。ポートチャンネル 1 インターフェイス ( e3/1-2 ) の出力 キーワードのキャプチャを設定して下さい。インターフェイスは以前に記述されているように例 0 に、属します。

```

N7K1# att mo 3
Attaching to module 3 ...
To exit type 'exit', to abort type '$.'
module-3# el asic flanker instance 0
module-3(fln-elam)# layer2
module-3(fln-l2-elam)# trigger dbus ipv4 egress if source-ipv4-address 192.168.13.3
destination-ipv4-address 192.168.12.2
module-3(fln-l2-elam)# trigger rbus egress if trig

```

```

module-3(fln-l2-elam)# status
ELAM Slot 3 instance 0: L2 DBUS Configuration: trigger dbus ipv4 egress if
source-ipv4-address 192.168.13.3 destination-ipv4-address 192.168.12.2
L2 DBUS: Configured
ELAM Slot 3 instance 0: L2 RBUS Configuration: trigger rbus egress if trig
L2 RBUS: Configured

```

```

module-3(fln-l2-elam)# start
module-3(fln-l2-elam)# status
ELAM Slot 3 instance 0: L2 DBUS Configuration: trigger dbus ipv4 egress if
source-ipv4-address 192.168.13.3 destination-ipv4-address 192.168.12.2
L2 DBUS: Armed
ELAM Slot 3 instance 0: L2 RBUS Configuration: trigger rbus egress if trig
L2 RBUS: Armed

```

```

module-3(fln-l2-elam)# status
ELAM Slot 3 instance 0: L2 DBUS Configuration: trigger dbus ipv4 egress if
source-ipv4-address 192.168.13.3 destination-ipv4-address 192.168.12.2
L2 DBUS: Triggered
ELAM Slot 3 instance 0: L2 RBUS Configuration: trigger rbus egress if trig
L2 RBUS: Triggered
module-3(fln-l2-elam)#

```

```

module-3(fln-l2-elam)# show dbus | in seq
sequence-number : 0x8d v1 : 0x3

```

!--- The sequence number is the same.

```
module-3(fln-l2-elam)# show rbus | in seq
vl                : 0x0          sequence-number      : 0x8d
```

```
module-3(fln-l2-elam)# show dbus
cp = 0x1007db4c, buf = 0x1007db4c, end = 0x10089e9c
```

-----  
Flanker Instance 00 - Capture Buffer On L2 DBUS:

Status(0x0102), TriggerWord(0x000), SampleStored(0x005), CaptureBufferPointer(0x005)

```
is_l2_egress: 0x0000, data_size: 0x023
[000]: 48c22000 08210000 40020800 0cc01414 5800a000 00001a40 01030000 00000000
00000000 00000000 003931c8 842850f9 31c88428 50800000 02358000 00000000 00000000
00000000 00000000 00000000 00000000 00005000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00605406 81e05406 0100008f e0054600 00000000
```

Printing packet 0

-----  
L2 DBUS PRS MLH IPV4  
-----

label-count	: 0x0	mc	: 0x0
null-label-valid	: 0x0	null-label-exp	: 0x0
null-label-ttl	: 0x0	lbl0-vld	: 0x0
lbl0-eos	: 0x0	lbl0-lbl	: 0x0
lbl0-exp	: 0x0	lbl0-ttl	: 0x0
lbl1-exp	: 0x0	lbl1-ttl	: 0x0
ipv4	: 0x0	ipv6	: 0x0
l4-protocol	: 0x1	df	: 0x0
mf	: 0x0	frag	: 0x0
ttl	: 0xfe	l3-packet-length	: 0x54
option	: 0x0	tos	: 0x0
sup-eid	: 0x0	header-type	: 0x1
error	: 0x0	redirect	: 0x0
port-id	: 0x1	last-ethertype	: 0x800
l2-frame-type	: 0x0	da-type	: 0x0
packet-type	: 0x1	l2-length-check	: 0x0
ip-da-multicast	: 0x0	ip-multicast	: 0x0
ip-multicast-control	: 0x0	ids-check-fail	: 0x0
traceroute	: 0x0	outer-cos	: 0x0
inner-cos	: 0x0	vqi-valid	: 0x1
vqi	: 0x40	packet-length	: 0x66
<b>vlan</b>	<b>: 0xa</b>	<b>destination-index</b>	<b>: 0xa2c</b>
<b>source-index</b>	<b>: 0x50</b>	<b>bundle-port</b>	<b>: 0x0</b>
acos	: 0x0	outer-drop-eligibility	: 0x0
inner-drop-eligibility	: 0x0	sg-tag	: 0x0
rbh	: 0xd2	vsl-num	: 0x0
inband-flow-creation-deletion	: 0x0	ignore-qoso	: 0x0
ignore-qosi	: 0x0	ignore-aclo	: 0x0
ignore-acli	: 0x0	index-direct	: 0x0
no-stats	: 0x0	dont-forward	: 0x0
notify-index-learn	: 0x1	notify-new-learn	: 0x0
disable-new-learn	: 0x0	disable-index-learn	: 0x0
dont-learn	: 0x0	bpdu	: 0x0
ff	: 0x0	rf	: 0x1
ccc	: 0x4	l2	: 0x0
rdt	: 0x0	dft	: 0x0
dfst	: 0x0	status-ce-1q	: 0x0
status-is-1q	: 0x0	trill-encap	: 0x0
mim-valid	: 0x0	dtag-ttl	: 0x0

```

dtag-ftag          : 0x0          valid          : 0x1
erspan-kpa-valid   : 0x0          recir-shim-vxlan-src-peer-id: 0x0
vn-valid          : 0x0          source-vif     : 0x0
destination-vif    : 0x0          vn-p           : 0x0
sequence-number    : 0x8d         vl             : 0x3
inner-de-valid     : 0x0          de-cfi         : 0x0
second-inner-cos   : 0x0          tunnel-type    : 0x0
shim-valid         : 0x0
segment-id-valid   : 0x0          copp           : 0x0
dti-type-vpnid    : 0x0          segment-id     : 0x0
ib-length-bundle   : 0x0          mlh-type       : 0x5
ulh-type           : 0x6
source-ipv4-address: 192.168.13.3
destination-ipv4-address: 192.168.12.2
mim-destination-mac-address : 0000.0000.0000
mim-source-mac-address  : 0000.0000.0000
destination-mac-address : e4c7.2210.a143

```

source-mac-address : e4c7.2210.a142 示されているように、送信元および宛先インデックスは両方とも DBUS ( 入力キャプチャで示されている同じでない ) の一部です。

```

module-3(fln-12-elam)# show rbus
cp = 0x100a2548, buf = 0x100a2548, end = 0x100ae898

```

```

-----
Flanker Instance 00 - Capture Buffer On L2 RBUS:

```

```

Status(0x1102), TriggerWord(0x000), SampleStored(0x008),CaptureBufferPointer(0x000)

```

```

is_l2_egress: 0x0001, data_size: 0x018
[000]: 0048ea00 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 0c000000 00000000 04014008 00005000 00000000
00000726 3910850a 1b931c88 42850800 00000000 00000000 0000008d

```

```

Printing packet 0

```

```

-----
L2 RBUS EGRESS CONTENT
-----

```

```

pad          : 0x0          valid          : 0x1
trig         : 0x1          reserved       : 0x0
vn-tag-p     : 0x0          cbl-vlan-valid : 0x0
vft-hop-count : 0x0          vft-vsant     : 0x0
vft-up       : 0x0          vft-valid     : 0x0
copp         : 0x0          segment-id-valid : 0x0
segment-id-23 : 0x0          vsl-num       : 0x0
inner-cos    : 0x0          inner-drop-eligibility: 0x0
cos          : 0x0          drop-eligibility : 0x0
dce-mode     : 0x0          flood-to-bd    : 0x0
pt-bit-en    : 0x1          cpu-port      : 0x0
vlan-id      : 0xa          ip-tos        : 0x0
result-rbh   : 0x1          met-ptr       : 0x2000
packet-type  : 0x1          sg-tag        : 0x0
dtag-ftag    : 0x0          vdc           : 0x0
vn-tag-src-vif : 0x0          vn-tag-dst-vif : 0x0
vn-tag-l     : 0x0          dc3-tr        : 0x0
vl           : 0x0          sequence-number : 0x8d

```

```

destination-mac-valid: 0x0
source-mac-valid: 0x0
mim-destination-mac-address : 0000.0000.0000
destination-mac-address : e4c7.2210.a143
source-mac-address : e4c7.2210.a142

```

mim-source-mac-address : 0000.0000.0000 送信元および宛先 IP アドレスは入力 ELAM パケットキャプチャの後でデコードされるように正しいです、;ただし、方向は、リターントラフ



ックとして入力 ELAM と比較されたときの反対にキャプチャ されず 確定的に あります。

10. VLAN 10 がそれを渡るトラフィックを転送するかどうか判別するために N7K1 のポート チャンネル 1 のモジュール 3 があるようにコロラド州ベースのロジック ( CBL ) を確認して下 さい。 CBL は物理インターフェイスごとの基づいたロジックです、従ってないポートチャ ンネル番号 N7K1 のポート チャンネル 1 のメンバー インターフェイス番号を入力する必要 があります。 次の出力では、VLAN 10 がそれを予想通り転送することがわかります。

CBL はハードウェア内のポートの Spanning Tree Protocol ( STP ) 状態を判別するために 使用されます。 Sup の VLAN があるように STP を確認するが、モジュールはトラフィッ クをブロックしますときインターフェイスがフォーワーディングを示すことは可能性のある です。 注: メンバー インターフェイスの両方があるように CBL をそれぞれ確認して下さい ( e3/1 および e3/2 )。 module-3# **show hardware internal mac port 1 table cbl vlan**

```
-----  
|                                     INGRESS                                     |  
| Disabled State   | 0,2-9,11-4031,4036-4095 |  
| Forwarding State | 1,10,4032-4035         |  
| Blocked State   |                          |  
| Learning State  |                          |  
-----  
|                                     EGRESS                                     |  
| Disabled State   | 0,2-9,11-4031,4036-4095 |  
| Forwarding State | 1,10,4032-4035         |  
| Blocked State   |                          |  
| Learning State  |                          |  
-----
```

注: 前の

コマンドはポート チャンネル 1 のためです ( モジュール 3 は e3/1 にあります )。 module-3# **show hardware internal mac port 2 table cbl vlan**

```
-----  
|                                     INGRESS                                     |  
| Disabled State   | 0,2-9,11-4031,4036-4095 |  
| Forwarding State | 1,10,4032-4035         |  
| Blocked State   |                          |  
| Learning State  |                          |  
-----  
|                                     EGRESS                                     |  
| Disabled State   | 0,2-9,11-4031,4036-4095 |  
| Forwarding State | 1,10,4032-4035         |  
| Blocked State   |                          |  
| Learning State  |                          |  
-----
```

注: 同様

に、このコマンドはポート チャンネル 2 ( e3/2 ) があるように CBL を確認します。

## ブレイクアウトケーブルで F3 シリーズ モジュールの ELAM を解決して下さい

ブレイクアウトケーブルが接続される時 F3 シリーズ モジュールのための ELAM プロシージャ は規則的なモジュールポートの ELAM 手順と異なりません。 ただしインターフェイスがブレイク アウトケーブルから届けば、フロントパネル数にインデックスを変換する試みの間にポート イン デックス マネージャ ( PIXM ) の確認に関していくつかの変更があります。

このセクション全体の例のために使用するトポロジはここにあります:

```
module-3# show hardware internal mac port 2 table cbl vlan
```

```
-----  
|                                     INGRESS                                     |  
| Disabled State   | 0,2-9,11-4031,4036-4095 |  
| Forwarding State | 1,10,4032-4035          |  
| Blocked State   |                          |  
| Learning State  |                          |  
-----  
|                                     EGRESS                                     |  
| Disabled State   | 0,2-9,11-4031,4036-4095 |  
| Forwarding State | 1,10,4032-4035          |  
| Blocked State   |                          |  
| Learning State  |                          |  
-----
```

この例に関しては、ブレイクアウトケーブルは4つの10 Gigabitポートに40ギガビットポートを分けるイーサネットインターフェイス3/8に接続されます。要求されたコンフィギュレーションは参照用にこのセクションで提供されます。

```
N7K3(config)# interface breakout module 3 port 8 map 10g-4x
```

```
N7K3(config)# show interface brief
```

```
---SNIP---
```

```
-----  
Ethernet      VLAN    Type Mode   Status Reason  Speed  Port  
Interface                                           Ch #  
-----  
Eth3/7        --      eth  routed up      none   40G(D) --  
Eth3/8/1      1       eth  trunk  up      none   10G(D) 2
```

```
!--- From 3/8/1 to 3/8/4.
```

```
Eth3/8/2      1       eth  trunk  up      none   10G(D) 2  
Eth3/8/3      1       eth  trunk  up      none   10G(D) 2  
Eth3/8/4      1       eth  trunk  up      none   10G(D) 2
```

前の出力では、イーサネットインターフェイス3/7が今でも40ギガビットポートであることがわかります;ただし、イーサネットインターフェイス3/8は個別に設定することができる4つの10 Gigabitポートに現在壊れています、:

```
N7K3# show run interface e3/8/1 - 4
```

```
!Command: show running-config interface Ethernet3/8/1-4
```

```
!Time: Mon May 4 01:46:28 2015
```

```
version 6.2(8a)
```

```
interface Ethernet3/8/1  
  switchport  
  switchport mode trunk  
  switchport trunk allowed vlan 10,20  
  no shutdown
```

```
interface Ethernet3/8/2  
  switchport  
  switchport mode trunk  
  switchport trunk allowed vlan 30,40  
  no shutdown
```

```
interface Ethernet3/8/3  
  switchport
```

```
switchport mode trunk
switchport trunk allowed vlan 50
no shutdown
```

```
interface Ethernet3/8/4
switchport
switchport mode trunk
no shutdown
```

N7K3 Switched Virtual Interface (SVI) 20 IP アドレスからのパケットキャプチャを始めて下さい (192.168.20.3) 4500 SVI 20 IP アドレスに (192.168.20.1)。パケットは 4500 への出力に N7K3 でキャプチャされ、4500 から N7K3 に応答は送られます。

前のセクションに記述されているように、トリガーを加えるためにフランカー例のナレッジを持たなければなりません。この出力はモジュール 3 の添付ファイルを示したものです:

```
N7K3# attach module 3
Attaching to module 3 ...
To exit type 'exit', to abort type '$.'
```

```
module-3# show hardware internal dev
dev-port-map dev-version
module-3# show hardware internal dev-port-map
```

```
-----
CARD_TYPE:      12 port 40G
>Front Panel ports:12
-----
```

```
-----
Device name           Dev role           Abbr num_inst:
-----
>Flanker Eth Mac Driver DEV_ETHERNET_MAC   MAC_0  6
>Flanker Fwd Driver     DEV_LAYER_2_LOOKUP L2LKP  6
>Flanker Xbar Driver     DEV_XBAR_INTF      XBAR_INTF 6
>Flanker Queue Driver   DEV_QUEUEING       QUEUE    6
>Sacramento Xbar ASIC   DEV_SWITCH_FABRIC  SWICHF  1
>Flanker L3 Driver      DEV_LAYER_3_LOOKUP L3LKP  6
>EDC                    DEV_PHY            PHYS    2
-----
```

```
+-----+
+-----+++FRONT PANEL PORT TO ASIC INSTANCE MAP+++-----+
+-----+
```

FP port	PHYS	MAC_0	L2LKP	L3LKP	QUEUE	SWICHF
1		0	0	0	0	0
2		0	0	0	0	0
3		1	1	1	1	0
4		1	1	1	1	0
5	0	2	2	2	2	0
6	0	2	2	2	2	0
7	1	3	3	3	3	0
8	1	3	<b>3</b>	3	3	0

!--- The port 8 L2LKP column shows a value of 3.

9		4	4	4	4	0
10		4	4	4	4	0
11		5	5	5	5	0
12		5	5	5	5	0

```
+-----+
+-----+
```

この出力では、例を知っているのでポート 8 はフランカー例 3 に送信元および宛先 IP アドレスによって、トリガーを置くことができます。N7K3 から 4500 に ping 要求をキャプチャするので、それは出力 ELAM です。

```
N7K3# attach module 3
Attaching to module 3 ...
To exit type 'exit', to abort type '$.'
```

```
module-3# show hardware internal dev
dev-port-map dev-version
module-3# show hardware internal dev-port-map
```

```
-----
CARD_TYPE:          12 port 40G
```

```
>Front Panel ports:12
```

```
-----
Device name          Dev role          Abbr num_inst:
-----
>Flanker Eth Mac Driver DEV_ETHERNET_MAC   MAC_0  6
>Flanker Fwd Driver    DEV_LAYER_2_LOOKUP L2LKP  6
>Flanker Xbar Driver   DEV_XBAR_INTF      XBAR_INTF 6
>Flanker Queue Driver  DEV_QUEUEING       QUEUE     6
>Sacramento Xbar ASIC  DEV_SWITCH_FABRIC  SWICHF   1
>Flanker L3 Driver     DEV_LAYER_3_LOOKUP L3LKP   6
>EDC                   DEV_PHY            PHYS     2
```

```
+-----+
+-----+++FRONT PANEL PORT TO ASIC INSTANCE MAP+++-----+
+-----+
```

FP port	PHYS	MAC_0	L2LKP	L3LKP	QUEUE	SWICHF
1		0	0	0	0	0
2		0	0	0	0	0
3		1	1	1	1	0
4		1	1	1	1	0
5	0	2	2	2	2	0
6	0	2	2	2	2	0
7	1	3	3	3	3	0
8	1	3	<b>3</b>	3	3	0

!--- The port 8 L2LKP column shows a value of 3.

9		4	4	4	4	0
10		4	4	4	4	0
11		5	5	5	5	0
12		5	5	5	5	0

```
+-----+
+-----+
PING は N7K3 から 4500 への始められます:
```

```
N7K3# ping 192.168.20.1
PING 192.168.20.1 (192.168.20.1): 56 data bytes
36 bytes from 192.168.20.3: Destination Host Unreachable
Request 0 timed out
64 bytes from 192.168.20.1: icmp_seq=1 ttl=254 time=6.49 ms
64 bytes from 192.168.20.1: icmp_seq=2 ttl=254 time=6.518 ms
64 bytes from 192.168.20.1: icmp_seq=3 ttl=254 time=7.936 ms
64 bytes from 192.168.20.1: icmp_seq=4 ttl=254 time=7.945 ms
```

```
--- 192.168.20.1 ping statistics ---
5 packets transmitted, 4 packets received, 20.00% packet loss
round-trip min/avg/max = 6.49/7.222/7.945 ms
```

ELAM ステータスはここにあります:

```
module-3(fln-l2-elam)# status
ELAM Slot 3 instance 3: L2 DBUS Configuration: trigger dbus ipv4 egress if
source-ipv4-address 192.168.20.3 destination-ipv4-address 192.168.20.1
L2 DBUS: Triggered
ELAM Slot 3 instance 3: L2 RBUS Configuration: trigger rbus egress if trig
```

L2 RBUS: Triggered

シーケンス番号が同じであることを確認して下さい:

```
module-3(fln-12-elam)# show dbus | in seq
sequence-number      : 0x27          vl          : 0x3
module-3(fln-12-elam)# show rbus | in seq
vl                   : 0x0          sequence-number : 0x27
```

シーケンス番号は同じです。この場合 DBUS および RBUS 情報をチェックできます:

```
module-3(fln-12-elam)# show dbus
cp = 0x1011033c, buf = 0x1011033c, end = 0x1011c68c
-----
Flanker Instance 03 - Capture Buffer On L2 DBUS:

Status(0x0102), TriggerWord(0x000), SampleStored(0x004),CaptureBufferPointer(0x004)

is_l2_egress: 0x0000, data_size: 0x023
[000]: 4c1ea000 20a10000 40021040 0cc02801 04080000 00000000 08100000 00000000
00000000 00000000 003c1fc1 8732dff9 31c88428 51000000 009d8000 00000000 00000000
00000000 00000000 00000000 00000000 00005000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 0060540a 01e0540a 0080008f f0054608 00000000
```

Printing packet 0

```
-----
L2 DBUS PRS MLH IPV4
-----
label-count          : 0x0          mc          : 0x0
null-label-valid    : 0x0          null-label-exp : 0x0
null-label-ttl      : 0x0          lb10-vld     : 0x0
lb10-eos            : 0x0          lb10-lbl     : 0x0
lb10-exp            : 0x0          lb10-ttl     : 0x0
lb11-exp            : 0x0          lb11-ttl     : 0x0
ipv4                : 0x0          ipv6         : 0x0
l4-protocol         : 0x1          df           : 0x0
mf                  : 0x0          frag        : 0x0
ttl                 : 0xff         l3-packet-length : 0x54
option              : 0x0          tos         : 0x0
sup-eid             : 0x1          header-type  : 0x0
error               : 0x0          redirect    : 0x0
port-id             : 0x5          last-ethertype : 0x800
l2-frame-type       : 0x0          da-type     : 0x0
packet-type         : 0x1          l2-length-check : 0x0
ip-da-multicast     : 0x0          ip-multicast : 0x0
ip-multicast-control: 0x0          ids-check-fail : 0x0
traceroute          : 0x0          outer-cos   : 0x0
inner-cos           : 0x0          vqi-valid   : 0x1
vqi                 : 0x82         packet-length : 0x66
vlan                : 0x14         destination-index : 0x82
source-index         : 0x400        bundle-port       : 0x0
acos                : 0x0          outer-drop-eligibility: 0x0
inner-drop-eligibility: 0x0          sg-tag      : 0x0
rbh                  : 0x0          vs1-num     : 0x0
inband-flow-creation-deletion: 0x0          ignore-qoso : 0x0
ignore-qosi         : 0x0          ignore-aclo : 0x0
ignore-acli         : 0x0          index-direct : 0x1
no-stats            : 0x0          dont-forward : 0x0
notify-index-learn  : 0x0          notify-new-learn : 0x0
disable-new-learn   : 0x0          disable-index-learn : 0x0
dont-learn          : 0x1          bpdu        : 0x0
ff                  : 0x0          rf          : 0x0
ccc                  : 0x0          l2          : 0x0
rdt                  : 0x0          dft         : 0x0
```

```

dfst : 0x0 status-ce-1q : 0x0
status-is-1q : 0x0 trill-encap : 0x0
mim-valid : 0x0 dtag-ttl : 0x0
dtag-ftag : 0x0 valid : 0x1
erspan-kpa-valid : 0x0 recir-shim-vxlan-src-peer-id: 0x0
vn-valid : 0x0 source-vif : 0x0
destination-vif : 0x0 vn-p : 0x0
sequence-number : 0x27 vl : 0x3
inner-de-valid : 0x0 de-cfi : 0x0
second-inner-cos : 0x0 tunnel-type : 0x0
shim-valid : 0x0
segment-id-valid : 0x0 copp : 0x0
dti-type-vpnid : 0x0 segment-id : 0x0
ib-length-bundle : 0x0 mlh-type : 0x5
ulh-type : 0x6
source-ipv4-address: 192.168.20.3
destination-ipv4-address: 192.168.20.1
mim-destination-mac-address : 0000.0000.0000
mim-source-mac-address : 0000.0000.0000
destination-mac-address : f07f.061c.cb7f
source-mac-address : e4c7.2210.a144

```

```

module-3(fln-12-elam)#
module-3(fln-12-elam)#
module-3(fln-12-elam)#
module-3(fln-12-elam)# show rbus
cp = 0x10134d38, buf = 0x10134d38, end = 0x10141088

```

-----

Flanker Instance 03 - Capture Buffer On L2 RBUS:

Status(0x1102), TriggerWord(0x000), SampleStored(0x008),CaptureBufferPointer(0x000)

```

is_l2_egress: 0x0001, data_size: 0x018
[000]: 004c4780 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 0c001000 00000000 80028010 00009000 00000000
00000783 f830e65b fb931c88 42851000 00000000 00000000 00000027

```

Printing packet 0

-----

L2 RBUS EGRESS CONTENT

-----

```

pad : 0x0 valid : 0x1
trig : 0x1 reserved : 0x0
vn-tag-p : 0x1 cbl-vlan-valid : 0x0
vft-hop-count : 0x0 vft-vsan : 0x0
vft-up : 0x0 vft-valid : 0x0
copp : 0x0 segment-id-valid : 0x0
segment-id-23 : 0x0 vsl-num : 0x0
inner-cos : 0x0 inner-drop-eligibility: 0x0
cos : 0x0 drop-eligibility : 0x0
dce-mode : 0x0 flood-to-bd : 0x0
pt-bit-en : 0x20 cpu-port : 0x0
vlan-id : 0x14 ip-tos : 0x0
result-rbh : 0x2 met-ptr : 0x4000
packet-type : 0x1 sg-tag : 0x0
dtag-ftag : 0x0 vdc : 0x0
vn-tag-src-vif : 0x0 vn-tag-dst-vif : 0x0
vn-tag-l : 0x0 dc3-tr : 0x0
vl : 0x0 sequence-number : 0x27
destination-mac-valid: 0x0
source-mac-valid: 0x0
mim-destination-mac-address : 0000.0000.0000

```

```
destination-mac-address : f07f.061c.cb7f
source-mac-address : e4c7.2210.a144
mim-source-mac-address : 0000.0000.0000
```

フローを確認するためにフロントパネル ポートに宛先および調査資料索引を変換して下さい:

```
N7K3# show system internal pixm info ltl 0x400
0x0400 is in SUP In-band LTL range
```

この出力は調査資料索引を示したものです。それが Sup から N7K3 に来る ping が理由で正しいことを確認します。次の出力はどの割り当て VLAN 20 か N7K の 2 つのインターフェイスの 1 つである出力 インターフェイス ( e3/8/1 ) を示したものです。他のインターフェイスは STP による 4500 でブロックされる e3/8/4 です。

```
N7K3# show system internal pixm info ltl 0x82
0x0082 is in DCE/FC pool
```

Member info

```
-----
Type                LTL
-----
PHY_PORT            Eth3/8/1
FLOOD_W_FPOE        0x8039
FLOOD_W_FPOE        0x803f
```

N7K のブレイクアウトケーブルで作成されたポートのための CBL を確認して下さい。CBL をチェックするために、新興ポートすべてのためのハードウェア ポート数を持たなければなりません。

**注:** インターフェイス e3/8 はスイッチにありません。新興ポートだけ現われます。

```
N7K3# show interface e3/8
^
% Incomplete command at '^' marker.
N7K3#
```

ブレイクアウトケーブルが使用され、e3/8 インターフェイスがスイッチで非存在であるので、ハードウェア ポート数を得るために使用する計算は変更します。ブレイクアウトをサポートするあらゆるモジュールに関しては、ハードウェア ポート番号は異なっています。ポートがブレイクアウトをサポートするかどうかまずチェックする必要があります:

```
N7K3# show int e3/7 capabilities
Ethernet3/7
Model:                N7K-F312FQ-25
Type (SFP capable):   QSFP-40G-CR4
Speed:                10000,40000
Duplex:               full
---SNIP---
PFC capable:         yes
Breakout capable:     yes
```

示されているように、ポート e3/7 はブレイクアウトをサポートします、つまり帯域幅が 4 つの 10 Gigabit ポートに分けることができることを意味します。同様に、100 ギガビット ポートを備えている他の F3 シリーズ モジュールは 10 ギガビットの 10 のポートそれぞれ、か加入超過を用いる 40 ギガビットの 3 つのポートに分けることができます。これはモジュールによって決まります。

この例の F3 シリーズ モジュールが 40 ギガビット ポートを備え、各ポートが 4 つのポートそれぞれに分けることができるので各ポートのためのハードウェア ポート番号は 0-3、4-7、8-11...40-43、ゼロ ベース スケールの 44-47 です。最初の例のためのポートのブレイクアウトケーブルがある場合、ハードウェア ポート番号は 0、1、2 および 3 です。ブレイクアウトケーブル

がない場合、ハードウェアポート番号は0です(1つ、2つ、および3つは非アクティブではないです)。

親ポートが e3/8 であるので、ハードウェアポート数はブレイクアウトケーブルによって使用される場合ブレイクアウトケーブルなしで使用される、28、29、30、および31です。場合28であり。このコマンド出力はアクティブなハードウェアポート(基づくゼロ)を示します:

```
N7K3# show system internal ifindex info mod 3
```

```
Init DB dump follows:
```

```
module_num_bitmask = 0x3ffff
```

```
Slot:3, Proc:1, breakout_factor:0, sw_card_id:0, active_cfg_ports:, broken_fp_ports:
```

```
Slot:3, Proc:2, breakout_factor:4, sw_card_id:155, active_cfg_ports:0,4,8,12,16,20,24,28-32,36,40,44, broken_fp_ports:28
```

```
Lookup DB dump follows:
```

```
Slot:3, breakout_factor:4
```

4に今分割される壊れたポート ハードウェアポート数は28です、(28-32)。この場合モジュール3を接続し、ハードウェアのCBLをチェックできます:

```
N7K3# attach module 3
```

```
Attaching to module 3 ...
```

```
To exit type 'exit', to abort type '$.'
```

```
module-3#
```

F3シリーズモジュールはポート番号を1ベースのスケールに従ってフォーマットされているために予想します。従って、29、30、31、および32を入力する必要があります:

```
module-3# show hardware internal mac port ?
```

```
<1-96> Port number (1-based)
```

```
!--- This is context sensitive, so it helps to say the port number is 1-based.
```

イーサネットインターフェイス 3/8/1 のための実行コンフィギュレーションは VLAN FORWARDING 状態を確認するためにチェックし、ここにあります:

```
module-3# show hardware internal mac port ?
```

```
<1-96> Port number (1-based)
```

```
!--- This is context sensitive, so it helps to say the port number is 1-based.module-3# show
```

```
hardware internal mac port 29 table cbl vlan
```

```
-----
```

INGRESS	
Disabled State	0,2-9,11-19,21-4031,4036-4095
Forwarding State	10,20,4032-4035
Blocked State	1
Learning State	

```
-----
```

EGRESS	
Disabled State	0,2-9,11-19,21-4031,4036-4095
Forwarding State	10,20,4032-4035
Blocked State	1
Learning State	

```
-----
```

イーサネットインターフェイス 3/8/2 のための実行コンフィギュレーションは VLAN FORWARDING 状態を確認するためにチェックし、ここにあります:

```
module-3# show hardware internal mac port 29 table cbl vlan
```

```
-----
```



```

|                                     INGRESS                                     |
| Disabled State | 0,2-9,11-19,21-4031,4036-4095 |
| Forwarding State | 10,20,4032-4035 |
| Blocked State | 1 |
| Learning State | |

```

```

|                                     EGRESS                                     |
| Disabled State | 0,2-9,11-19,21-4031,4036-4095 |
| Forwarding State | 10,20,4032-4035 |
| Blocked State | 1 |
| Learning State | |

```

-----module-3# **show**

**hardware internal mac port 30 table cbl vlan**

```

|                                     INGRESS                                     |
| Disabled State | 0,2-29,31-39,41-4031,4036-4095 |
| Forwarding State | 30,40,4032-4035 |
| Blocked State | 1 |
| Learning State | |

```

```

|                                     EGRESS                                     |
| Disabled State | 0,2-29,31-39,41-4031,4036-4095 |
| Forwarding State | 30,40,4032-4035 |
| Blocked State | 1 |
| Learning State | |

```

イーサネットインターフェイス 3/8/3 のための実行コンフィギュレーションは VLAN FORWARDING 状態を確認するためにチェックし、ここにあります:

module-3# **show hardware internal mac port 30 table cbl vlan**

```

|                                     INGRESS                                     |
| Disabled State | 0,2-29,31-39,41-4031,4036-4095 |
| Forwarding State | 30,40,4032-4035 |
| Blocked State | 1 |
| Learning State | |

```

```

|                                     EGRESS                                     |
| Disabled State | 0,2-29,31-39,41-4031,4036-4095 |
| Forwarding State | 30,40,4032-4035 |
| Blocked State | 1 |
| Learning State | |

```

-----module-3# **show**

**hardware internal mac port 31 table cbl vlan**

```

|                                     INGRESS                                     |
| Disabled State | 0,2-49,51-4031,4036-4095 |
| Forwarding State | 50,4032-4035 |
| Blocked State | 1 |
| Learning State | |

```

```

|                                     EGRESS                                     |
| Disabled State | 0,2-49,51-4031,4036-4095 |
| Forwarding State | 50,4032-4035 |
| Blocked State | 1 |
| Learning State | |

```

イーサネットインターフェイス 3/8/4 のための実行コンフィギュレーションは VLAN

FORWARDING 状態を確認するためにチェックし、ここにあります (すべての構成済みのVLANが割り当てられます):

```
module-3# show hardware internal mac port 31 table cbl vlan
```

```
-----
```

INGRESS	
Disabled State	0,2-49,51-4031,4036-4095
Forwarding State	50,4032-4035
Blocked State	1
Learning State	

```
-----
```

EGRESS	
Disabled State	0,2-49,51-4031,4036-4095
Forwarding State	50,4032-4035
Blocked State	1
Learning State	

```
-----
```

-----module-3# show

```
hardware internal mac port 32 table cbl vlan
```

```
-----
```

INGRESS	
Disabled State	0,2-9,11-19,21-29,31-39,41-49,51-59,61-669,671-4031
Disabled State	4036-4095
Forwarding State	1,20,30,40,50,60,670,4032-4035
Blocked State	10
Learning State	

```
-----
```

EGRESS	
Disabled State	0,2-9,11-19,21-29,31-39,41-49,51-59,61-669,671-4031
Disabled State	4036-4095
Forwarding State	1,20,30,40,50,60,670,4032-4035
Blocked State	10
Learning State	

```
-----
```

CBL は正しい VLAN が転送されることを示します。

ハードウェア ポート番号を得るために `show hardware internal エラー モジュール <module number>` コマンドを使用できます。このコマンドは `show interface x/y` コマンド 出力に現われない内部ドロップをチェックする必要があるとき役立ちます。次に例を示します。

```
N7K2# show hardware internal errors module 3
```

```
---SNIP---
```

```
Instance:1
```

Cntr	Name	Value	Ports
3836	igr rx pl: cbl drops	0000000000000001	10 -
4636	igr rx pl: cbl drops	0000000000000001	14 -

```
Instance:2
```

Cntr	Name	Value	Ports
423	igr in upm: pkts with symbol/sequence error rcvd	00000000000000478	18 -
455	igr in upm: pkts with symbol/sequence error rcvd	00000000000000478	17 -
487	igr in upm: pkts with symbol/sequence error rcvd	00000000000000478	19 -
519	igr in upm: pkts with symbol/sequence error rcvd	00000000000000478	20 -

```
Instance:3
```

Cntr	Name	Value	Ports
------	------	-------	-------

423	igr in upm: pkts with symbol/sequence error rcvd	0000000000000745	26	-
455	igr in upm: pkts with symbol/sequence error rcvd	0000000000000745	25	-
487	igr in upm: pkts with symbol/sequence error rcvd	0000000000000745	27	-
519	igr in upm: pkts with symbol/sequence error rcvd	0000000000000745	28	-
550	igr in upm: pkts rcvd, with RCODE violation	0000359810913821	30	-
551	igr in upm: pkts with symbol/sequence error rcvd	0000425092490108	30	-
552	igr in upm: pkts with error	0000000000176136	30	-
582	igr in upm: pkts rcvd, with RCODE violation	0000000000292641	29	-
583	igr in upm: pkts with symbol/sequence error rcvd	0000000000114014	29	-
614	igr in upm: pkts rcvd, with RCODE violation	0000133362265995	31	-
615	igr in upm: pkts with symbol/sequence error rcvd	0000146701474013	31	-
616	igr in upm: pkts with error	0000000000157479	31	-
646	igr in upm: pkts rcvd, with RCODE violation	0000000002160959	32	-
647	igr in upm: pkts with symbol/sequence error rcvd	0000000003722562	32	-
648	igr in upm: pkts with error	0000000000000002	32	-