# Exploiting Inaccurate Branch History in Side-Channel Attacks

**Yuhui Zhu**[1,2], Alessandro Biondi[1]

[1]Scuola Superiore Sant'Anna, [2]Scuola IMT Alti Studi Lucca

Sant'Anna
School of Advanced Studies – Pisa

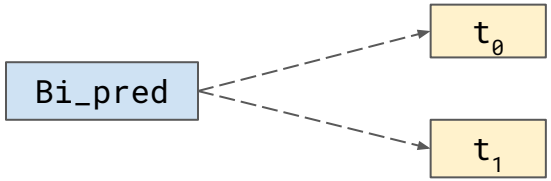
# TL;DR

1. Systematically analyze resource sharing and contention in modern branch prediction units;
2. Identify mechanisms that enable Branch History Buffer manipulation;
3. Present novel side-channel attacks: Spectre-BSE, Spectre-BHS, and BiasScope;
4. Demonstrate exploitable patterns via Chimera – an end-to-end eBPF attack.

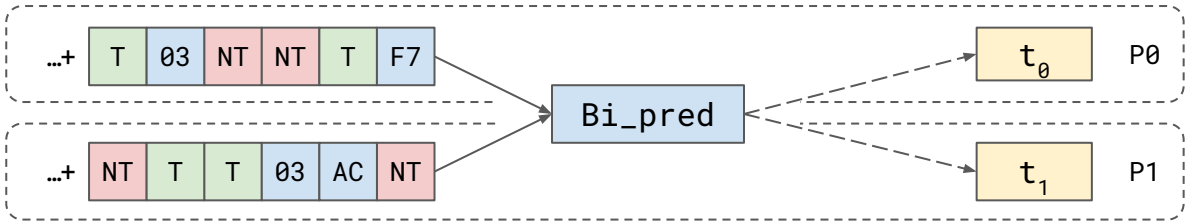
# Outline

1. **Background**
2. Bias-Free Branch Prediction (only Cortex-A72)
3. Branch History Speculation
4. Chimera Snippet and eBPF
5. Conclusion

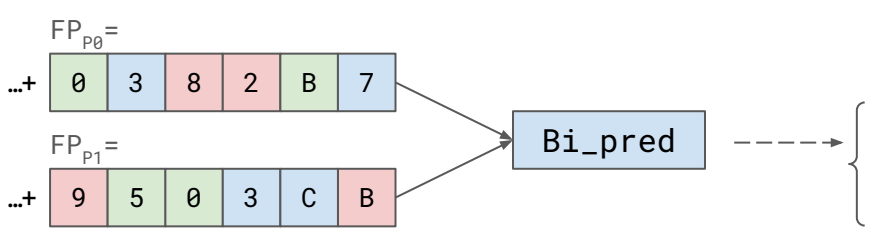
# Background: *Indirect* Branch Target Buffer



# Background: *Indirect* Branch Target Buffer



# Background: *Indirect* Branch Target Buffer




iBTB

| TAG | BHB | PC | TARGET |
|---|---|---|---|
| TAG$_{P0}$ | 82B7 | &Bi_pred | $t_0$ |
| TAG$_{P1}$ | 03CB | &Bi_pred | $t_1$ |
| *** | *** | &Bi_vul | t_leak |

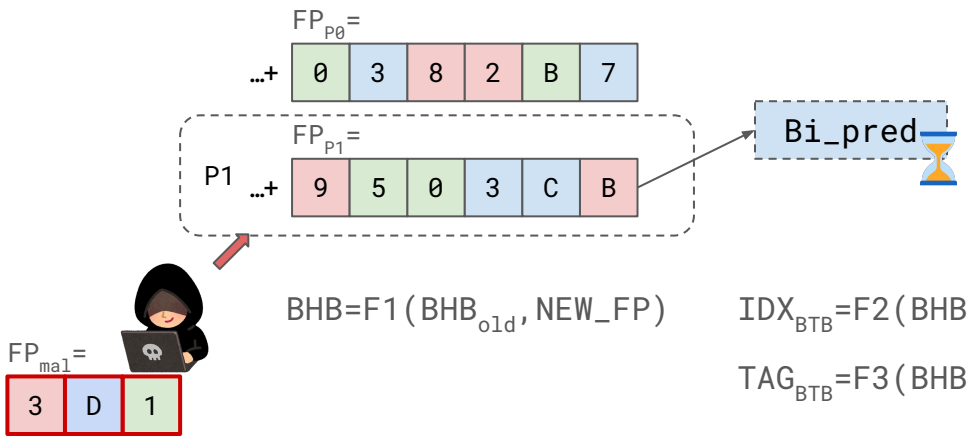
$BHB=F1(BHB_{old},NEW\_FP)$

$IDX_{BTB}=F2(BHB,PC)$

$TAG_{BTB}=F3(BHB,PC)$

# Background: Branch History Injection (2022)

- Crafts BHB values to induce BTB index collisions.



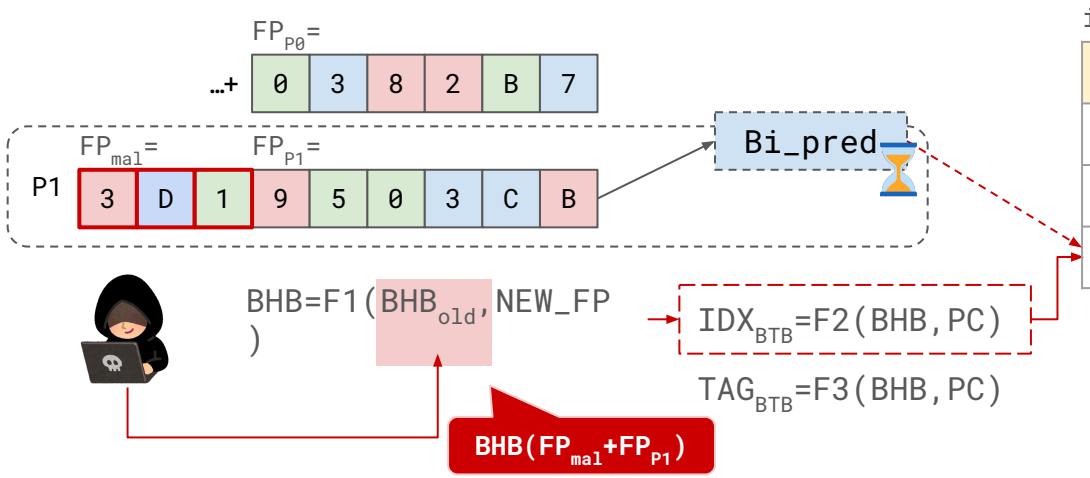

iBTB

| TAG | BHB | PC | TARGET |
|---|---|---|---|
| $TAG_{P0}$ | 82B7 | &Bi_pred | $t_0$ |
| $TAG_{P1}$ | 03CB | &Bi_pred | $t_1$ |
| *** | *** | &Bi_vul | t_leak |

E. Barberis, P. Frigo, M. Muench, H. Bos, and C. Giuffrida, “Branch history injection: On the effectiveness of hardware mitigations against Cross-Privilege spectre-v2 attacks,” in 31st USENIX security symposium (USENIX security 22), Boston, MA: USENIX Association, Aug. 2022, pp. 971–988.

# Background: Branch History Injection (2022)

- Crafts BHB values to induce BTB index collisions.



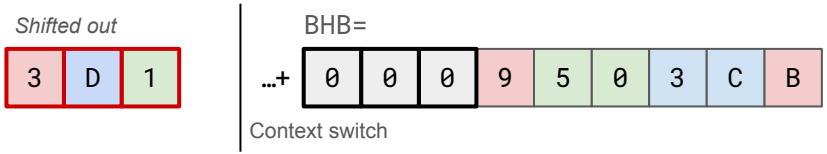

iBTB

| TAG | BHB | PC | TARGET |
|---|---|---|---|
| $TAG_{P0}$ | 82B7 | &Bi_pred | $t_0$ |
| $TAG_{P1}$ | 03CB | &Bi_pred | $t_1$ |
| *** | *** | &Bi_vul | t_leak |

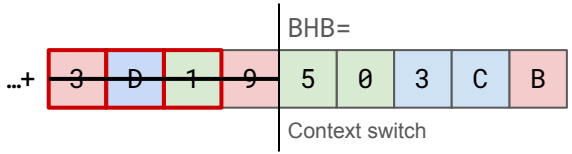
E. Barberis, P. Frigo, M. Muench, H. Bos, and C. Giuffrida, “Branch history injection: On the effectiveness of hardware mitigations against Cross-Privilege spectre-v2 attacks,” in 31st USENIX security symposium (USENIX security 22), Boston, MA: USENIX Association, Aug. 2022, pp. 971–988.

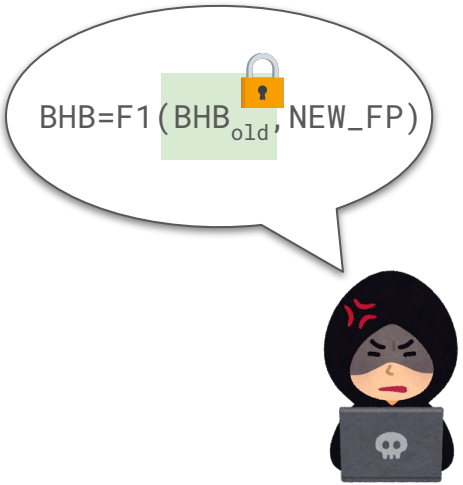
# Spectre-BHI Mitigations

- BHB sanitization: Arm BHB clearing sequence

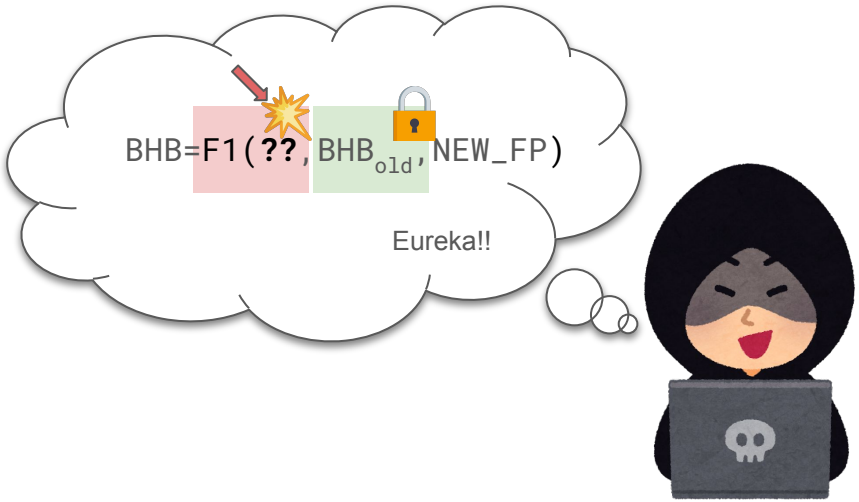



- BHB isolation: Intel BHI_NO



# Spectre-BHI Mitigations

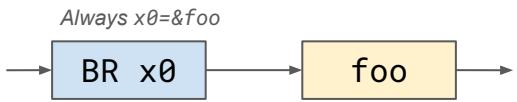


# Outline

1. Background
2. **Bias-Free Branch Prediction (only Cortex-A72)**
3. Branch History Speculation
4. Chimera Snippet and eBPF
5. Conclusion

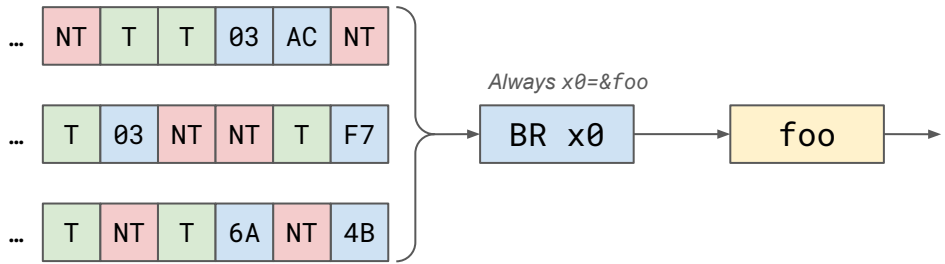
# Non-informative branches in BHB

- Branches with weak path correlation (e.g., biased branches) are less informative for BHB-based prediction.



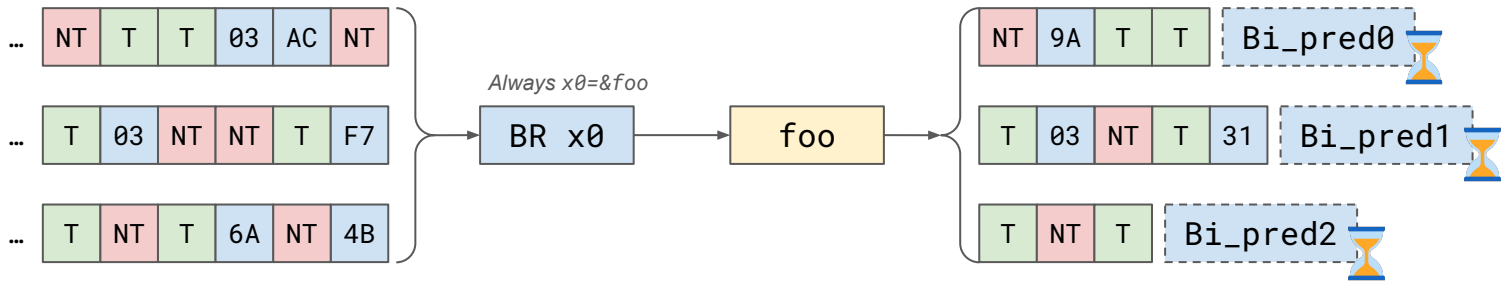
# Non-informative branches in BHB

- Branches with weak path correlation (e.g., biased branches) are less informative for BHB-based prediction.



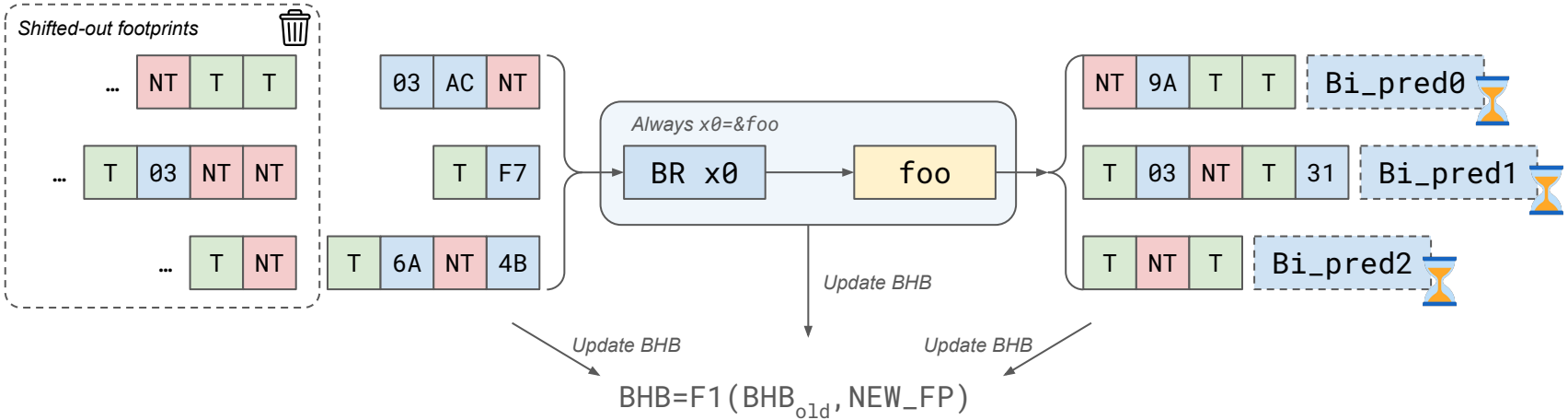
# Non-informative branches in BHB

- Branches with weak path correlation (e.g., biased branches) are less informative for BHB-based prediction.



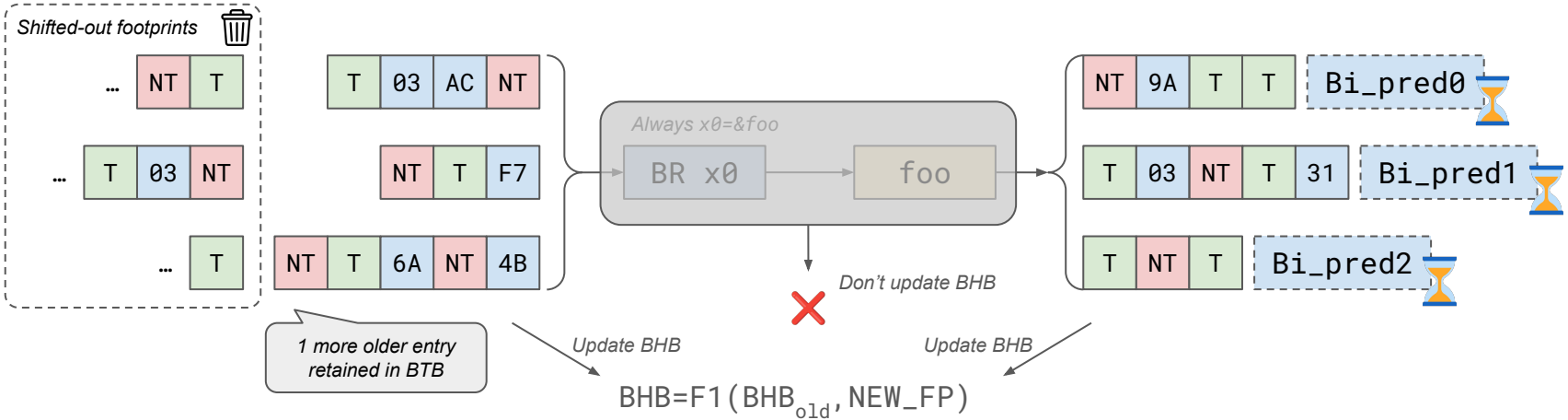
# Non-informative branches in BHB

- Branches with weak path correlation (e.g., biased branches) are less informative for BHB-based prediction.



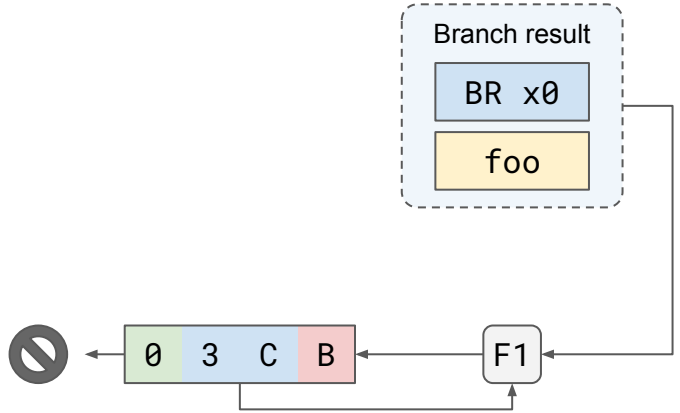
# Non-informative branches in BHB

- Branches with weak path correlation (e.g., biased branches) are less informative for BHB-based prediction.
- Filter out *biased branches* to retain more informative BHB entries.



# Bias-Free Branch Prediction

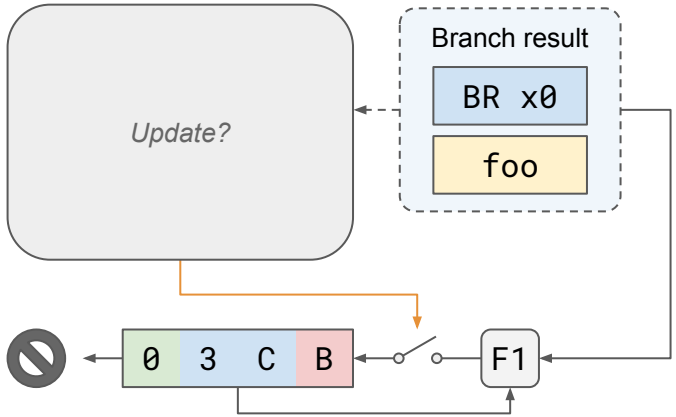
- Introduces a mechanism to selectively record branch footprints in the BHB.




$BHB=F1(BHB_{old}, NEW\_FP)$

# Bias-Free Branch Prediction

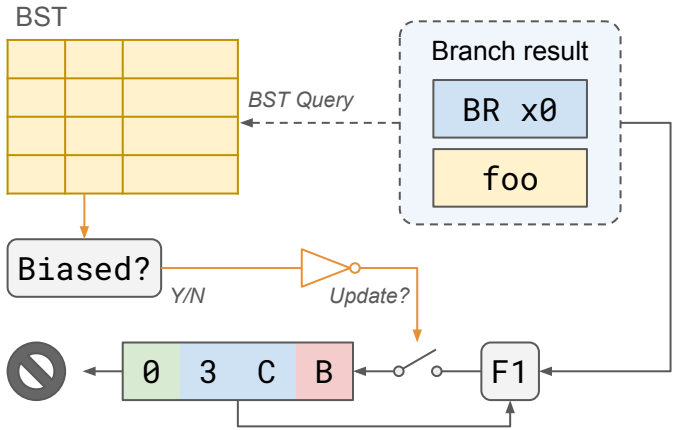
- Introduces a mechanism to selectively record branch footprints in the BHB.




$BHB=\mathbf{F1(??,}\ BHB_{old},\ NEW\_FP\mathbf{)}$

# Bias-Free Branch Prediction

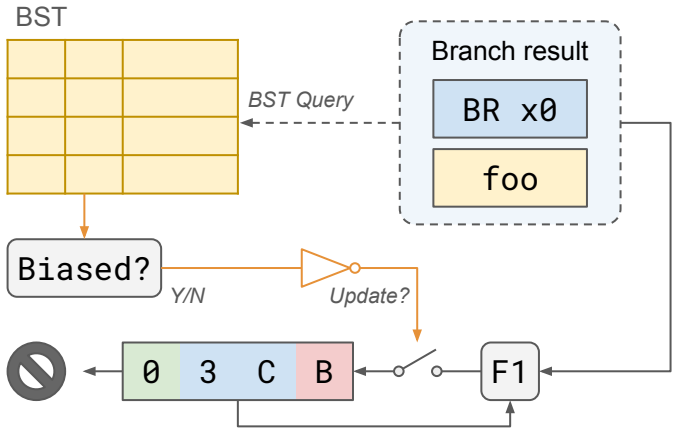
- Extra *Branch Status Table*.
- Only records the footprint of an indirect branch if the BST query returns approval.
- Fields:
  - Last branch outcome;
  - Bias status;
  - Tag.




$$BHB = \mathbf{F1}(\mathbf{BST}, BHB_{old}, NEW\_FP)$$

# Bias-Free Branch Prediction

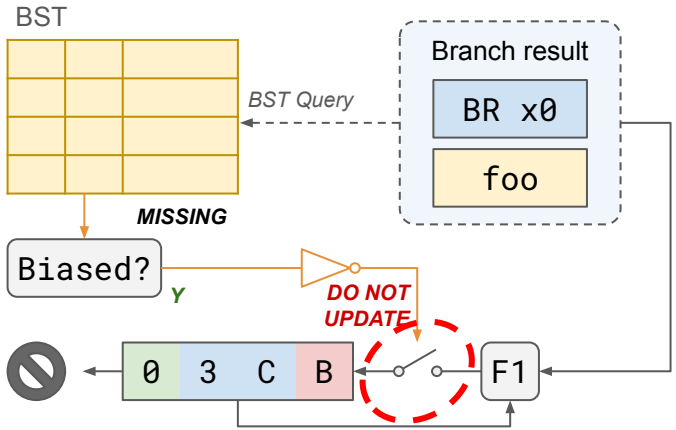
- When BHB Is Not Updated:
  a. BST miss: branch not seen before (default);
  b. BST hit, but:
     - Bias status = Yes, and
     - branch target matches the record.




$$BHB = \mathbf{F1}(\mathbf{BST}, BHB_{old}, NEW\_FP)$$

# Bias-Free Branch Prediction

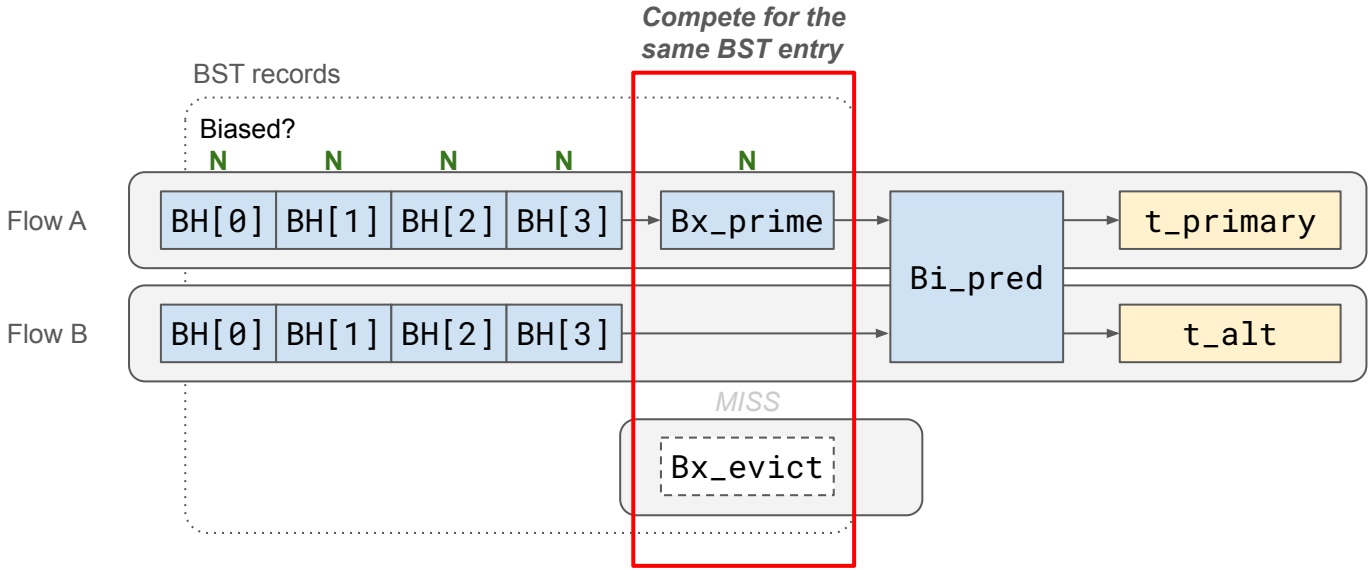
- When BHB Is Not Updated:
  a. BST miss: branch not seen before;
  b. BST hit, but:
     - Bias status = Yes, and
     - branch target matches the record.
- Evicting BST Records:
  a. Forces BST miss → branch treated as biased;
  b. Prevents BHB updates until the branch takes an alternative target.



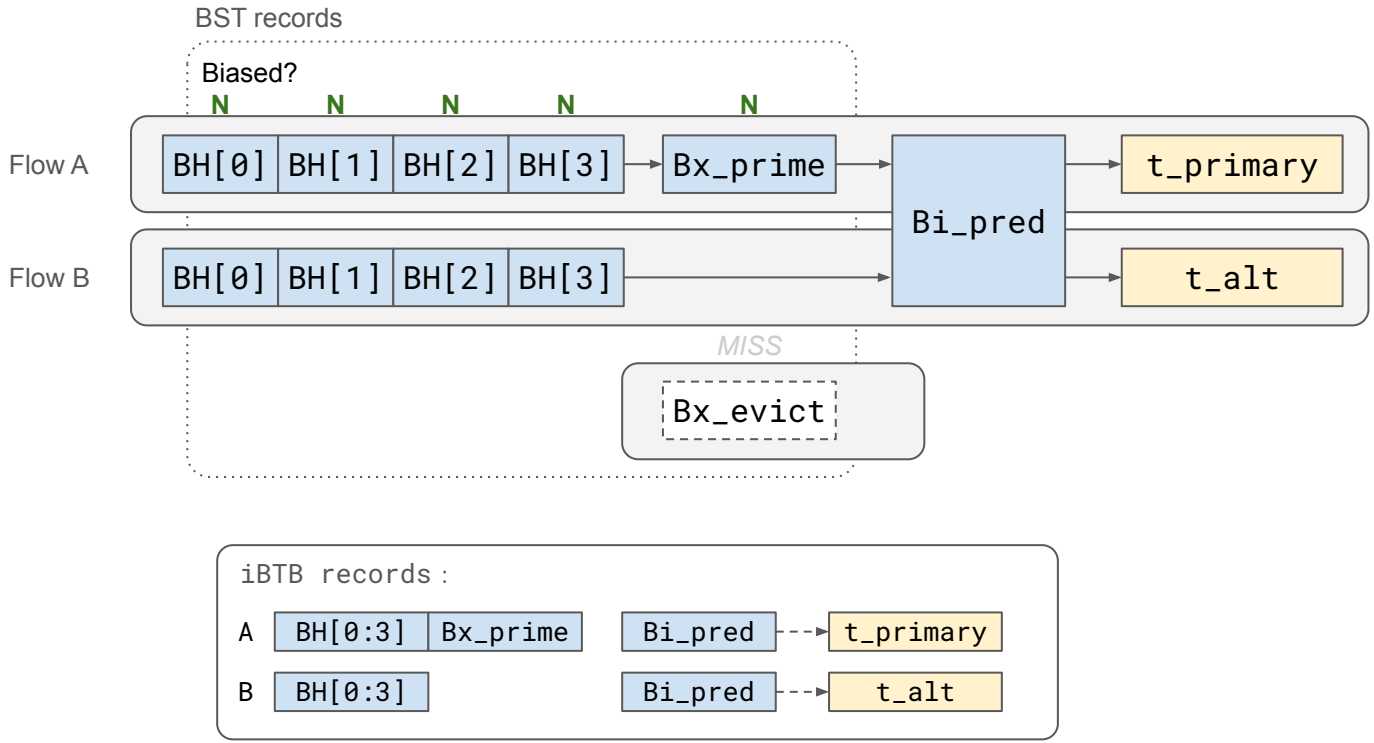

$BHB = \mathbf{F1}(\mathbf{BST}, BHB_{old}, NEW\_FP)$

# BST Eviction and BHB updating

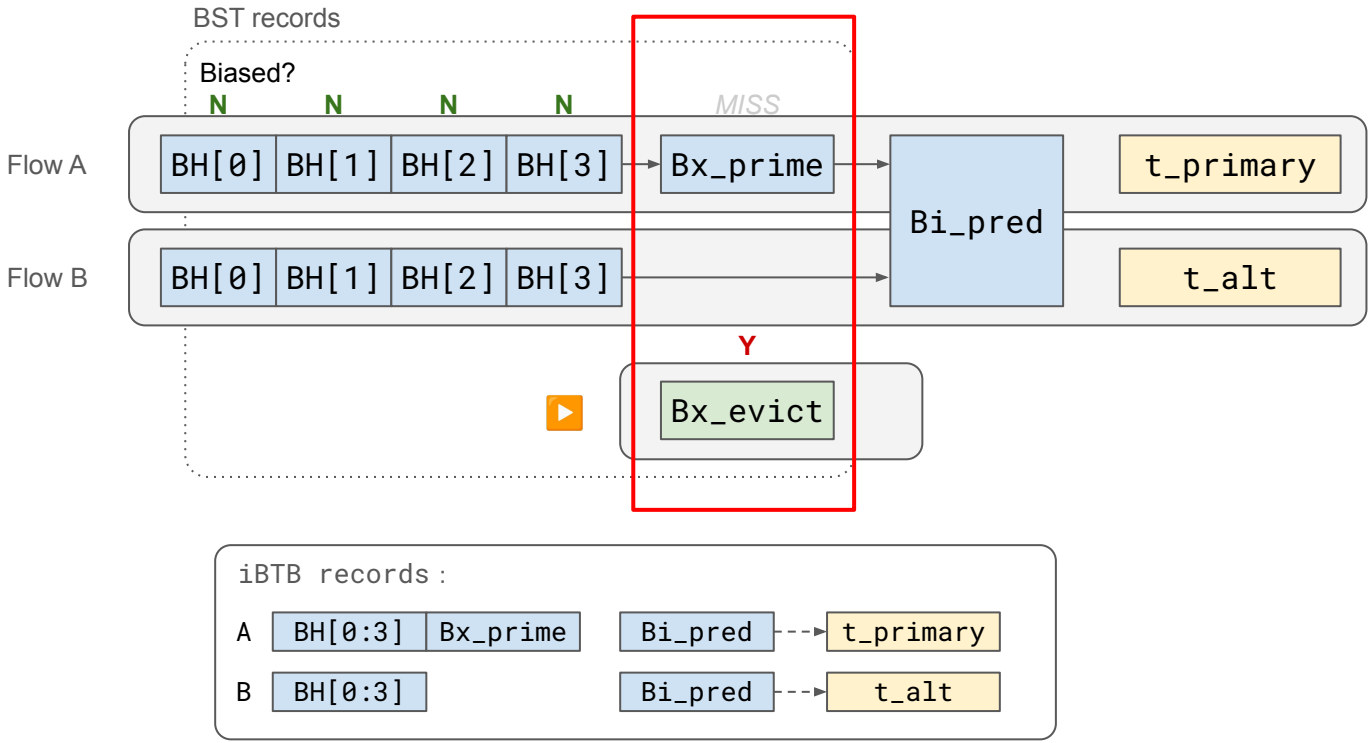
# BST Eviction and BHB updating



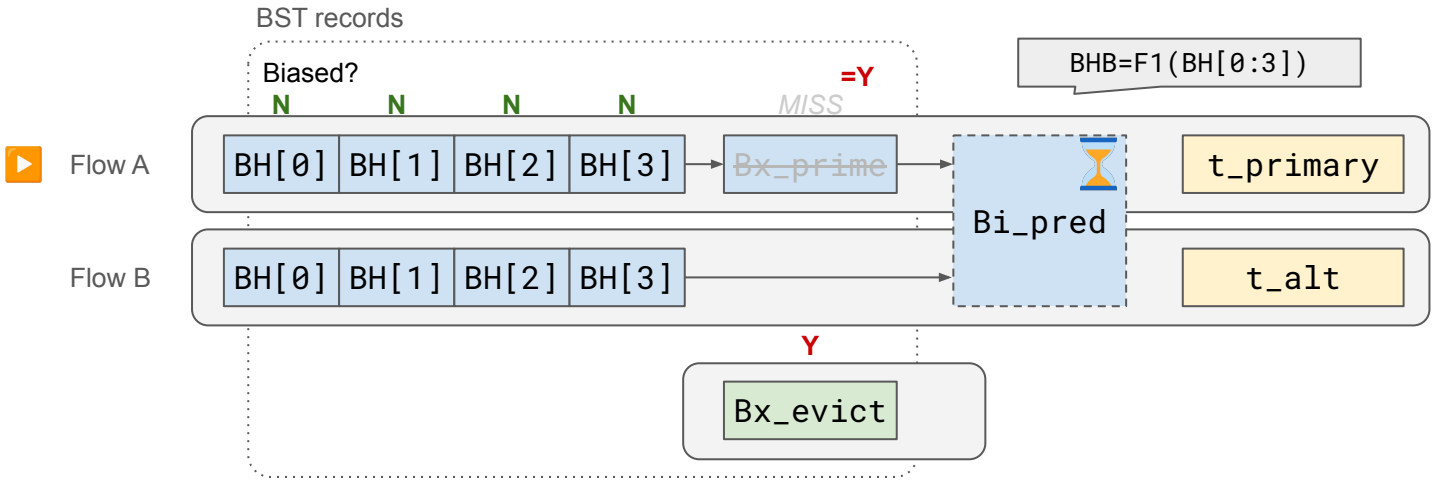
# BST Eviction and BHB updating

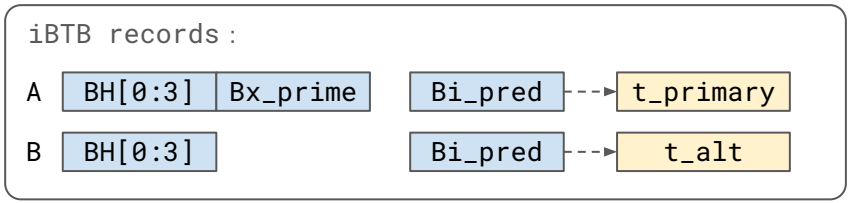


# BST Eviction and BHB updating

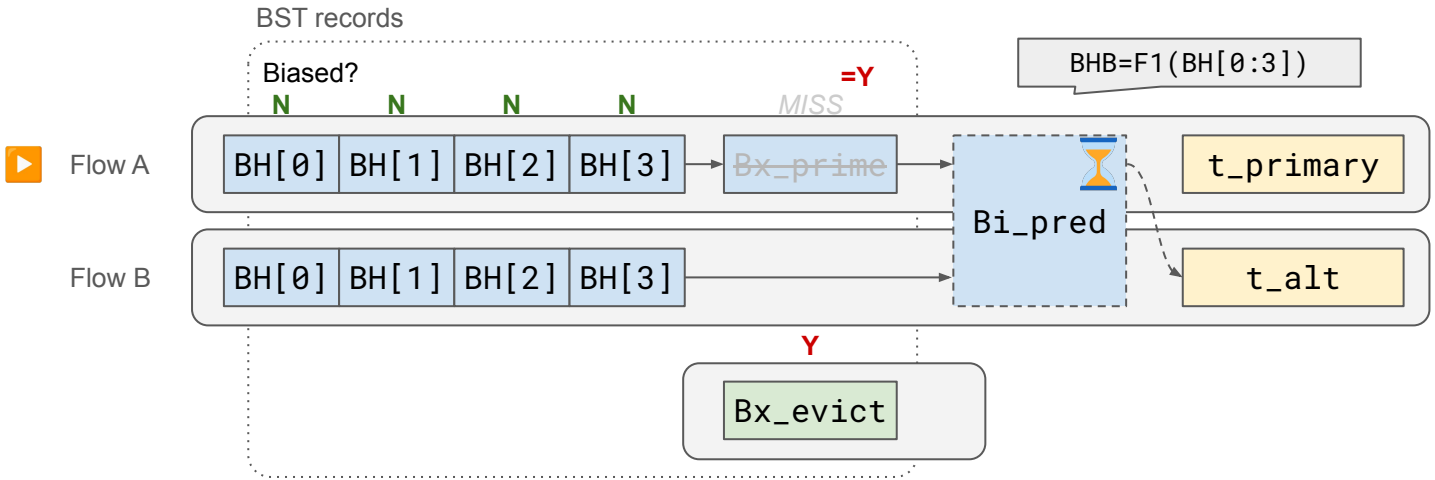


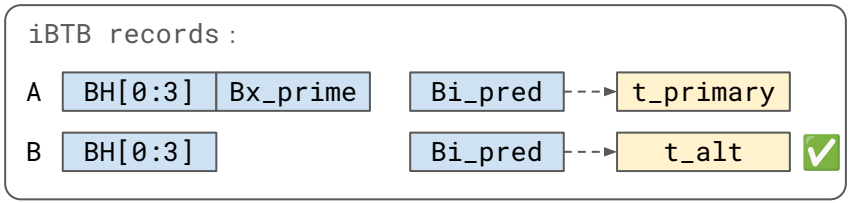
# BST Eviction and BHB updating



# BST Eviction and BHB updating



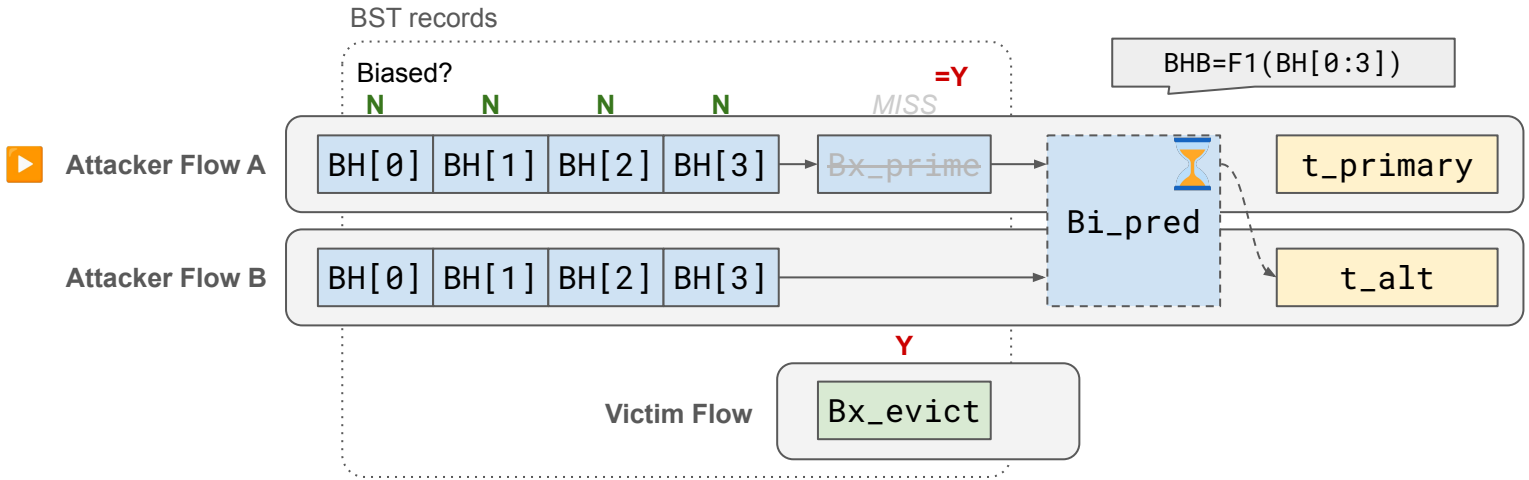



# BST Eviction and BHB updating

| Type | Mnemonics | Evict? |
|---|---|---|
| Indirect | `BR, BLR` | **Yes** |
| Conditional | `B.cond, TB(N)Z, CB(N)Z` | ***When Taken*** |
| Unconditional | `B, BL` | No |
| Return (Indirect) | `RET` | No |
| Other | `SVC` | No |

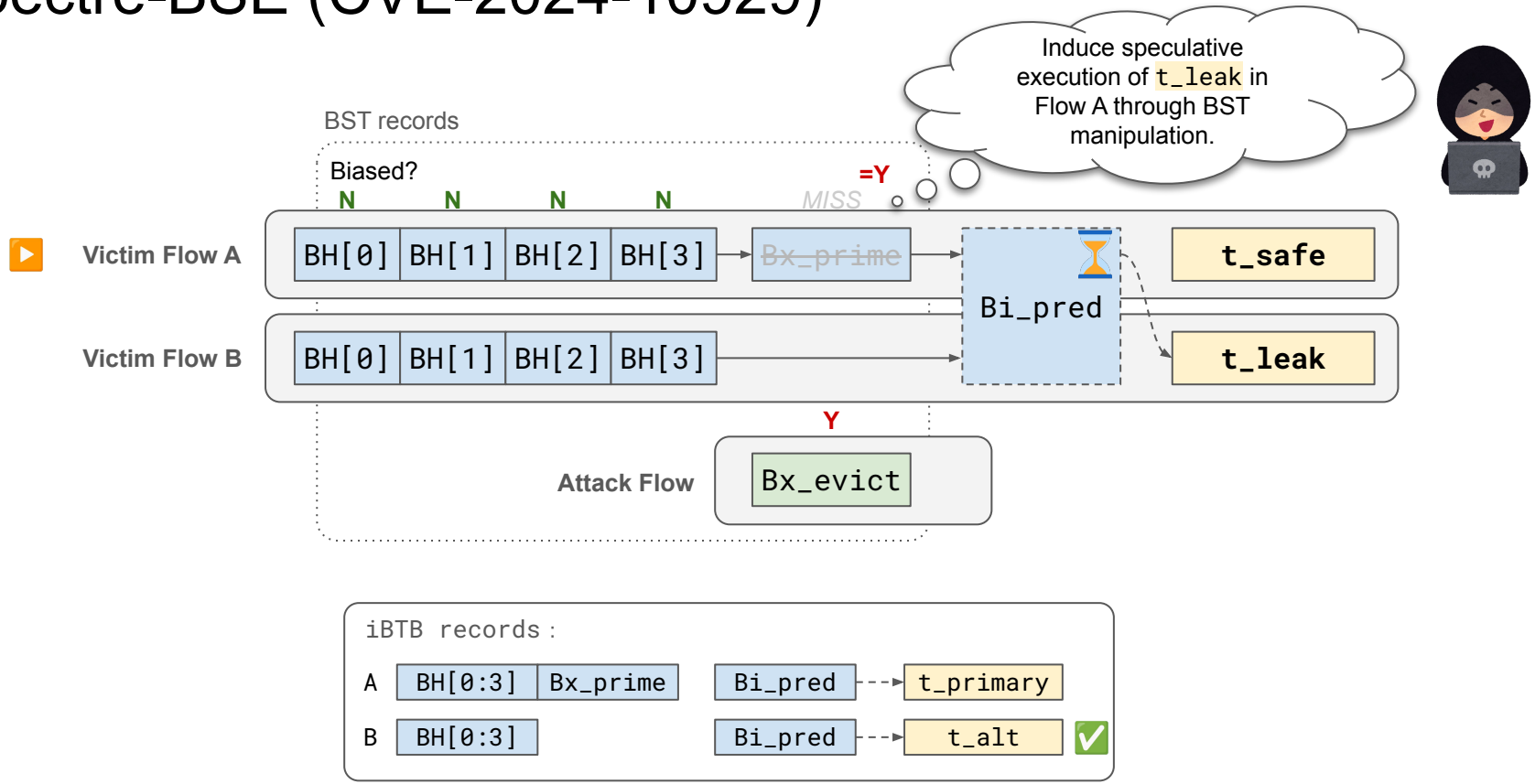
# BST Eviction and BHB updating

# BiasScope




- `t_primary` executed in Flow A → `Bx_evict` was not taken
- `t_alt` executed in Flow A → `Bx_evict` was taken

# Spectre-BSE (CVE-2024-10929)



# Existing mitigations on Cortex-A72

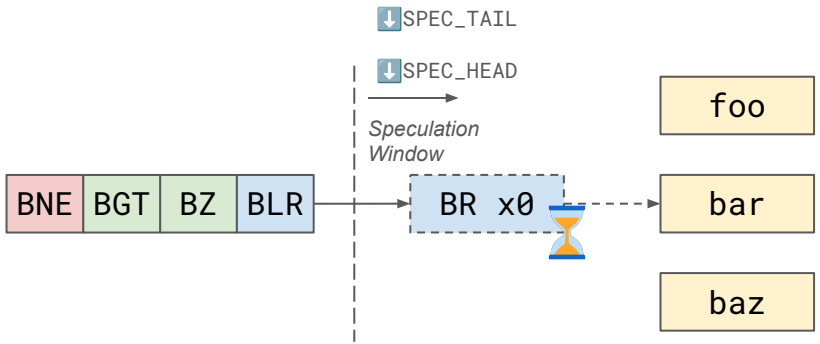
| Leakage | User -> User | Kernel -> User |
|---|---|---|
| Spectre mitigations | BPU flushing & BHB clearing | BHB clearing |
| BiasScope | **No** | **Yes** |
| Spectre-BSE | **No** | **Yes** |

# Outline

1. Background
2. Bias-Free Branch Prediction (only Cortex-A72)
3. **Branch History Speculation**
4. Chimera Snippet and eBPF
5. Conclusion

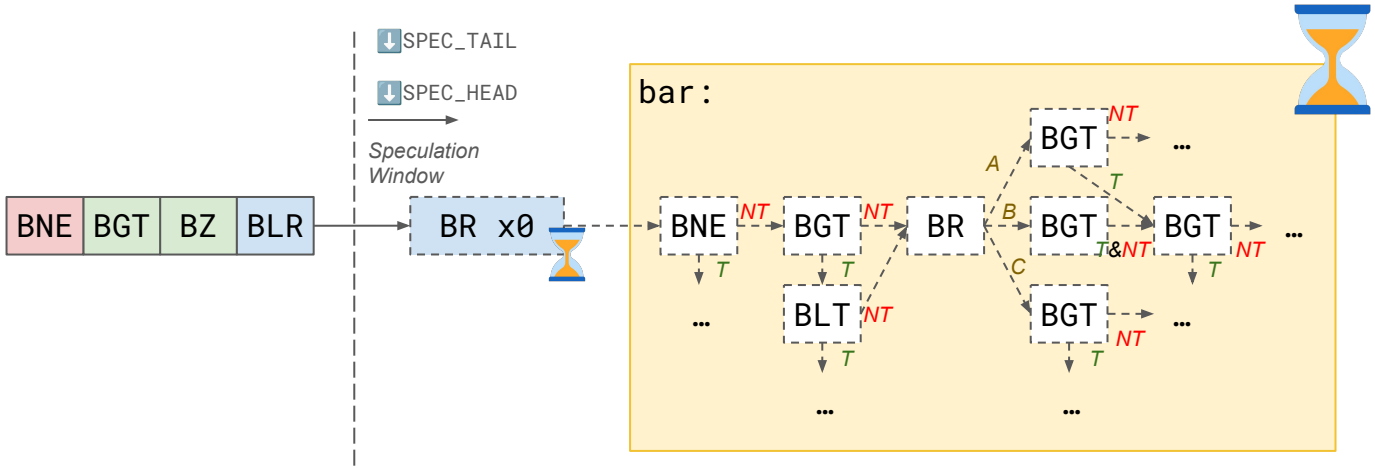
# Cascaded speculative execution

- Deeper, cascaded speculative execution prevents pipeline stalls.



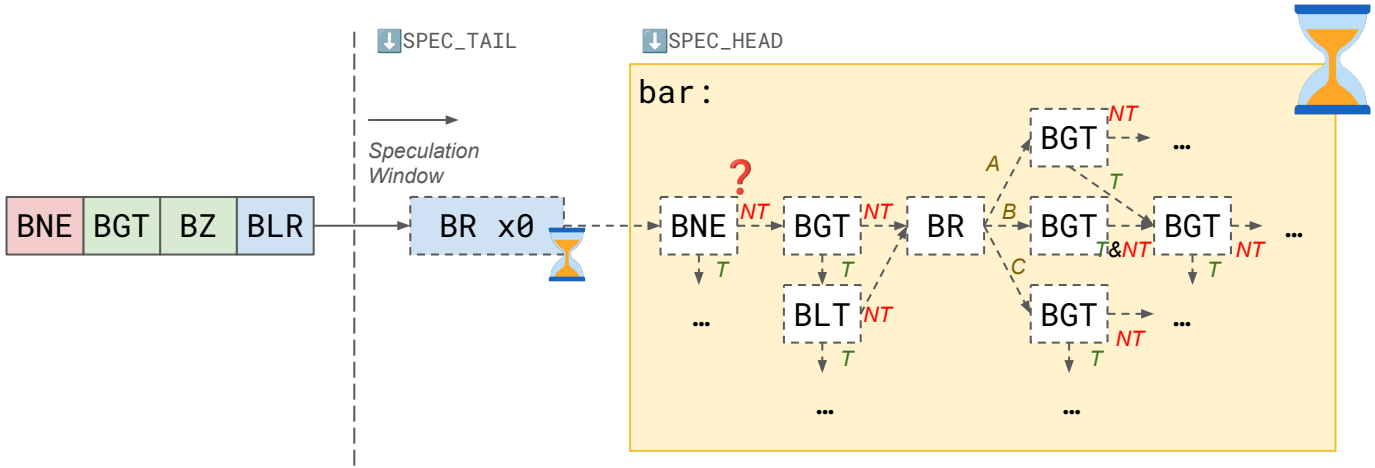
# Cascaded speculative execution

- Deeper, cascaded speculative execution prevents pipeline stalls.



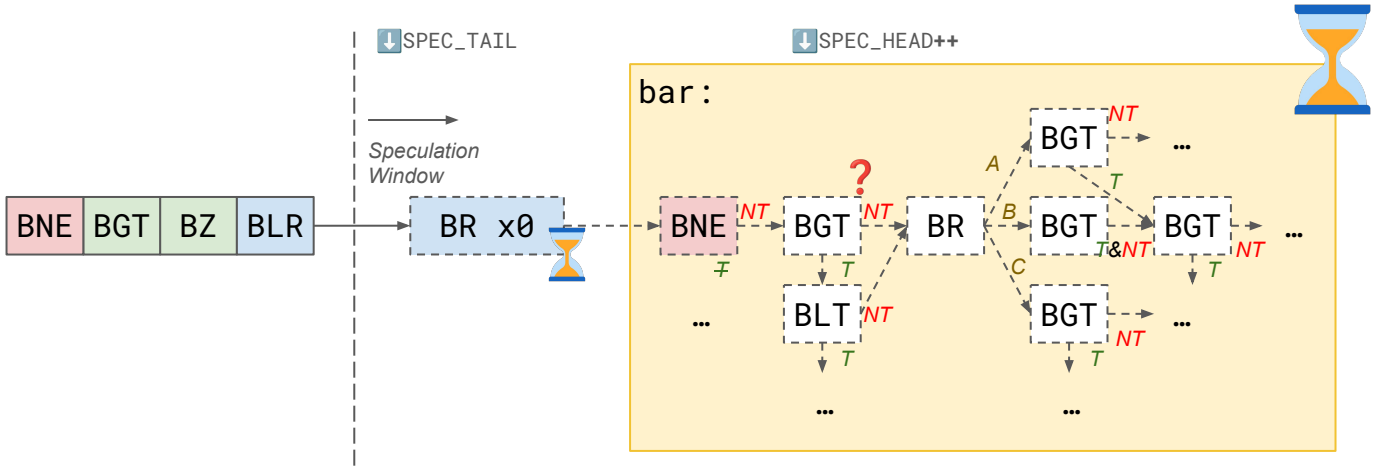
# Cascaded speculative execution

- Deeper, cascaded speculative execution prevents pipeline stalls.



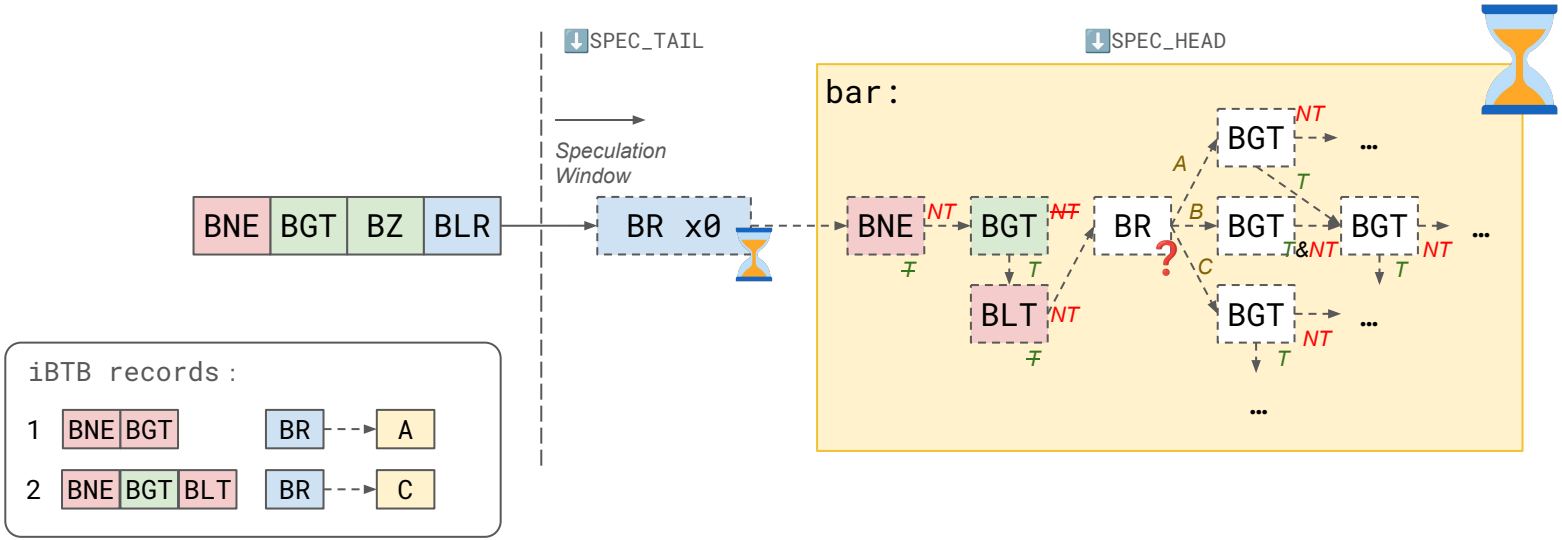
# Cascaded speculative execution

- Deeper, cascaded speculative execution prevents pipeline stalls.



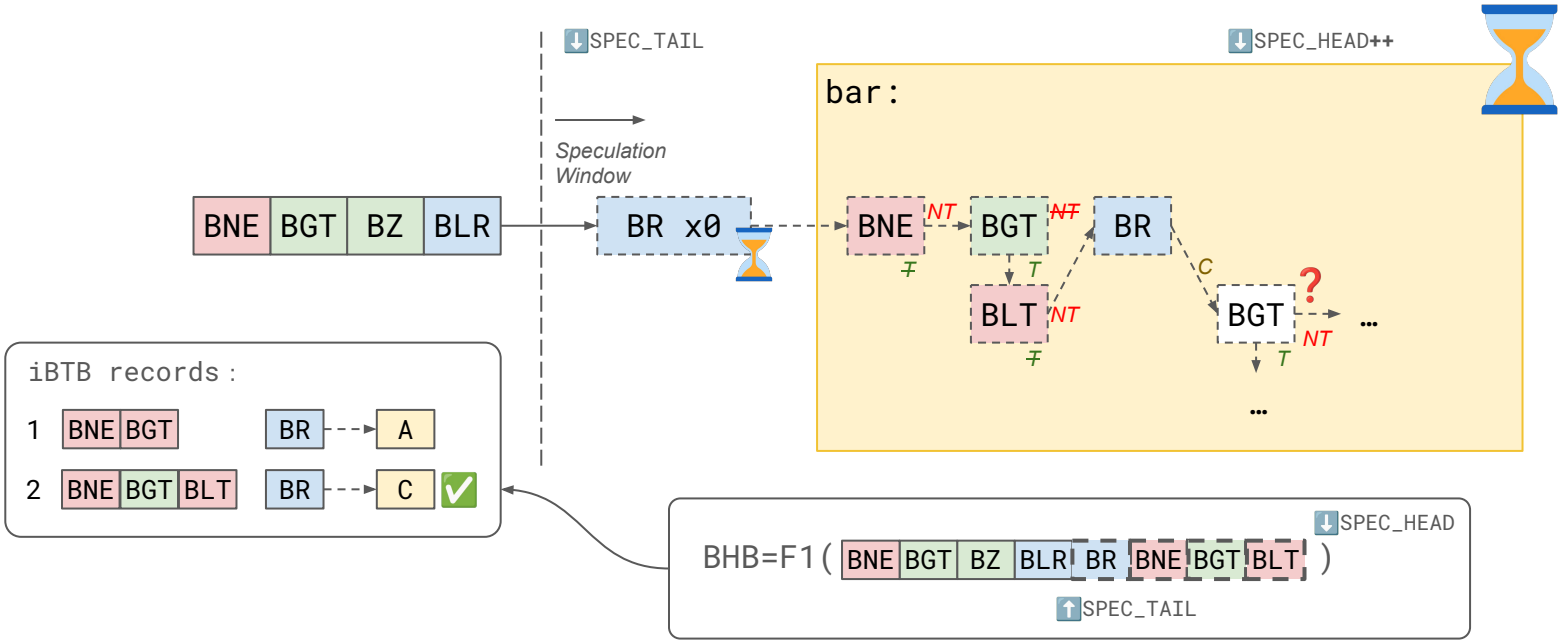
# Branch History Speculation

- Speculated branch outcomes should inform future predictions to maintain accuracy.



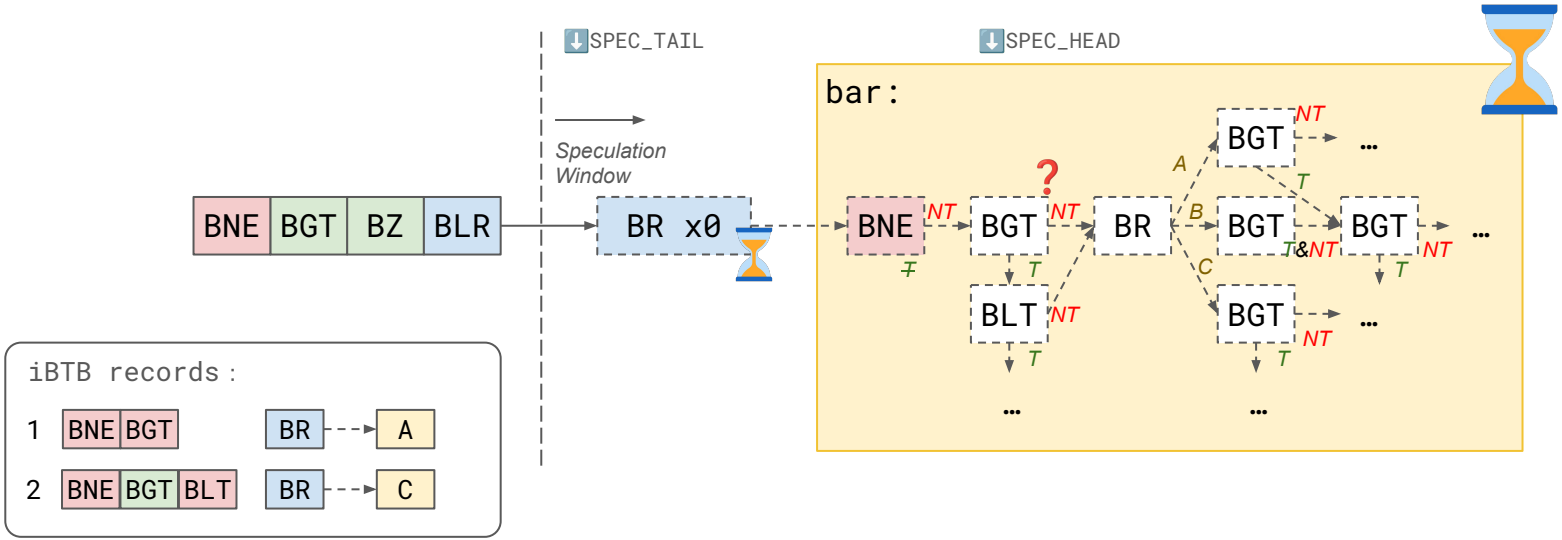
# Branch History Speculation

- Speculated branch outcomes should inform future predictions to maintain accuracy.



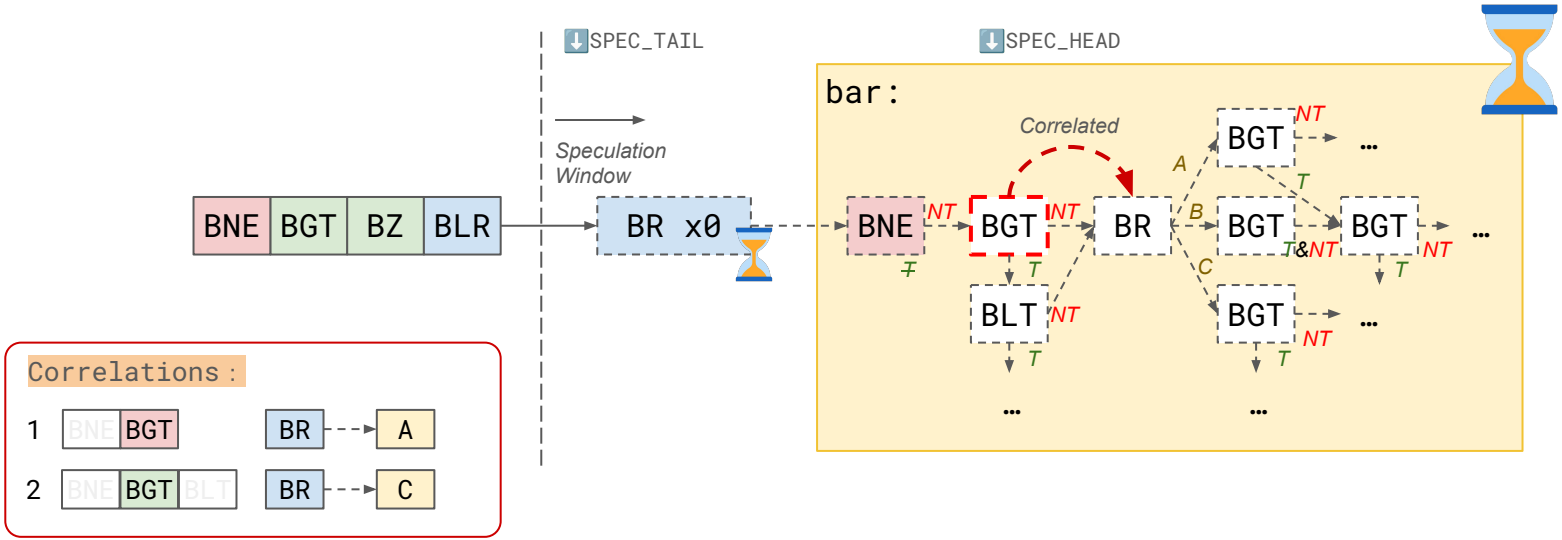
# Speculated but inaccurate branch history

1. Spectre-v1: Single mis-training affects future branch speculation.



# Speculated but inaccurate branch history

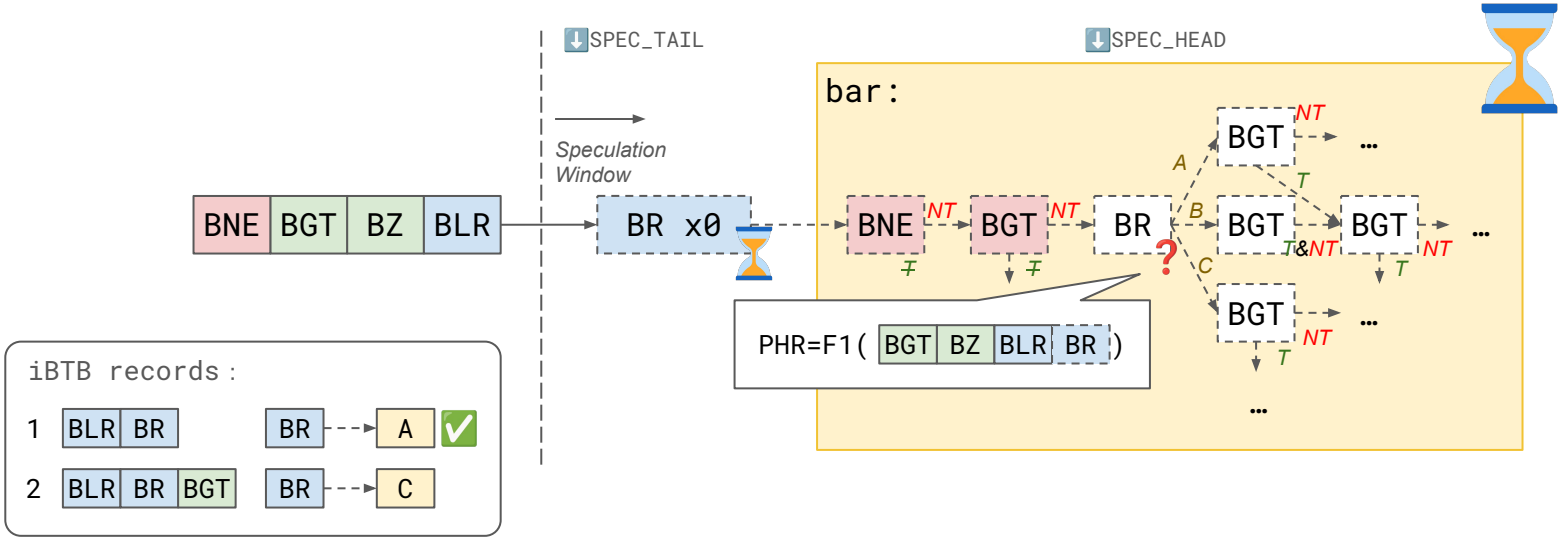
1. Spectre-v1: Single mis-training affects future branch speculation.



# Speculated but inaccurate branch history

2. Path-based BHB (PHR):

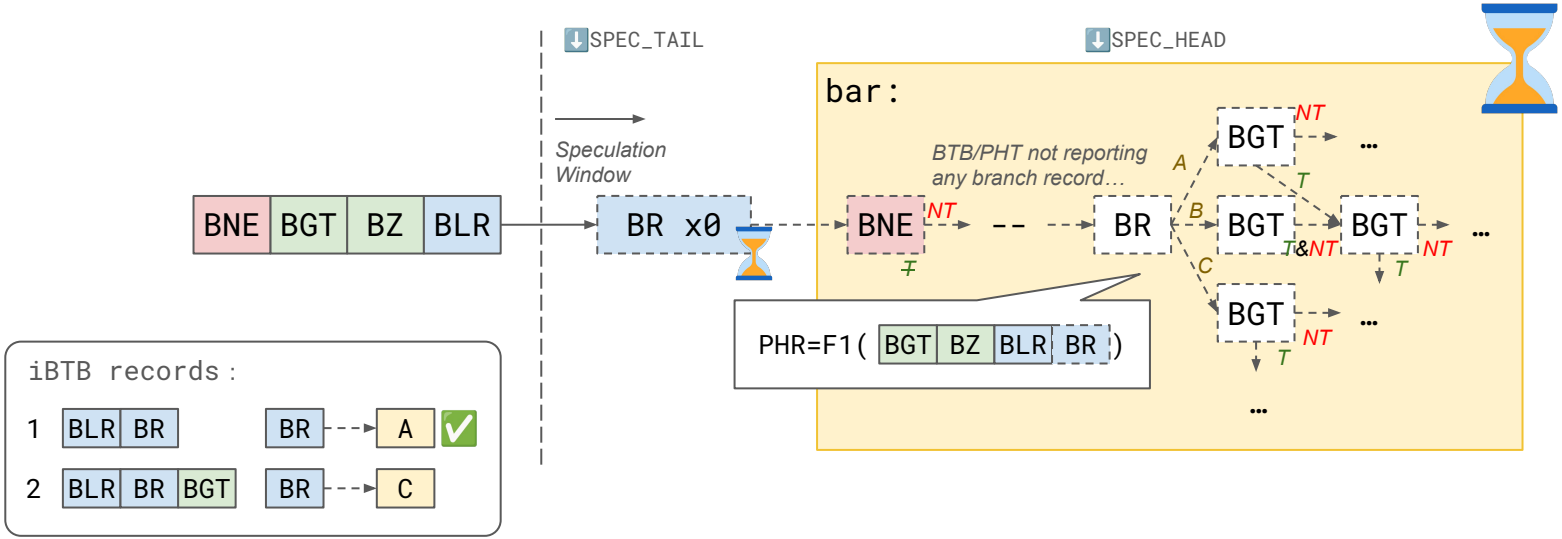
BTB/PHT eviction → BPU forgets branch → branch mis-interpreted as not-taken.



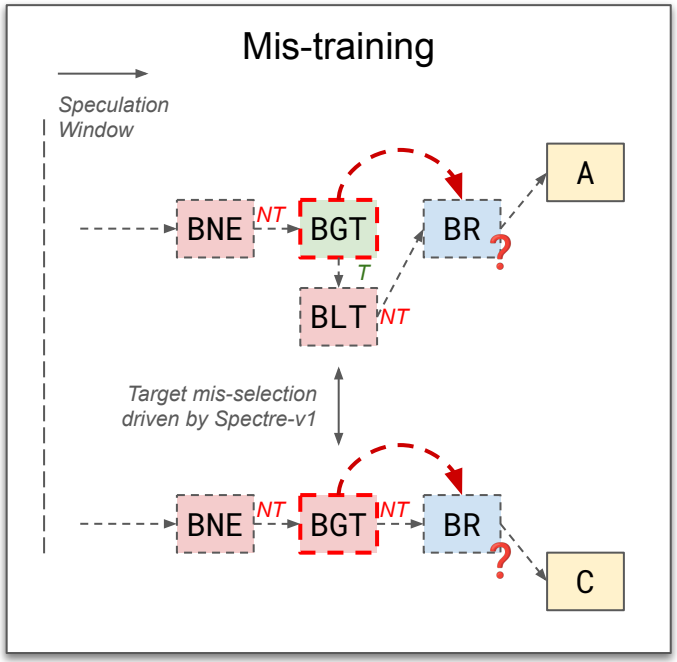
# Speculated but inaccurate branch history

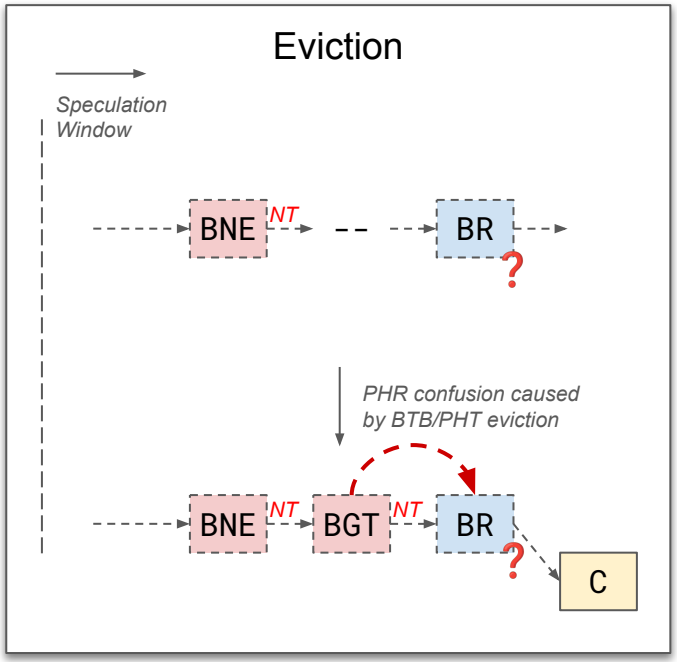
2. Path-based BHB (PHR):

BTB/PHT eviction → BPU forgets branch → branch mis-interpreted as not-taken.



# Spectre-BHS


Mis-training
Speculation Window
BNE
NT
BGT
T
BLT
NT
BR
A
Target mis-selection driven by Spectre-v1
BNE
NT
BGT
NT
BR
C


Eviction
Speculation Window
BNE
NT
BR
PHR confusion caused by BTB/PHT eviction
BNE
NT
BGT
NT
BR
C

# Spectre-BHS (cross-privilege)

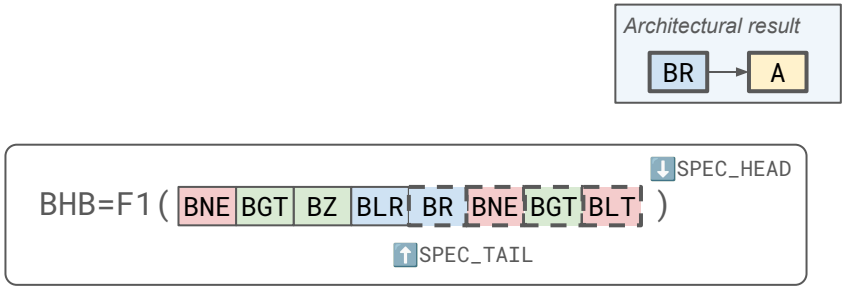
| | Mis-train | Eviction |
|---|---|---|
| Arm Cortex-A72 | **Yes** | Yes, but canonical BHB |
| Arm Cortex-A76/78 | **Yes** | **Yes** |
| AMD Zen4 | Only conditional branches | Only conditional branches, canonical BHB |
| Intel Gracemont / Redwood Cove / Crestmont | Only conditional branches | No |

# Outline

1. Background
2. Bias-Free Branch Prediction (only Cortex-A72)
3. Branch History Speculation
4. **Chimera Snippet and eBPF**
5. Conclusion

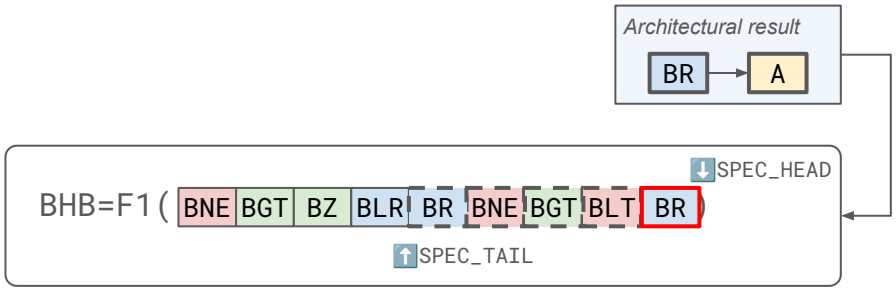
# Architectural footprint in speculated branch history

- BHB receives updates from architecturally committed branches and speculative predictions simultaneously.



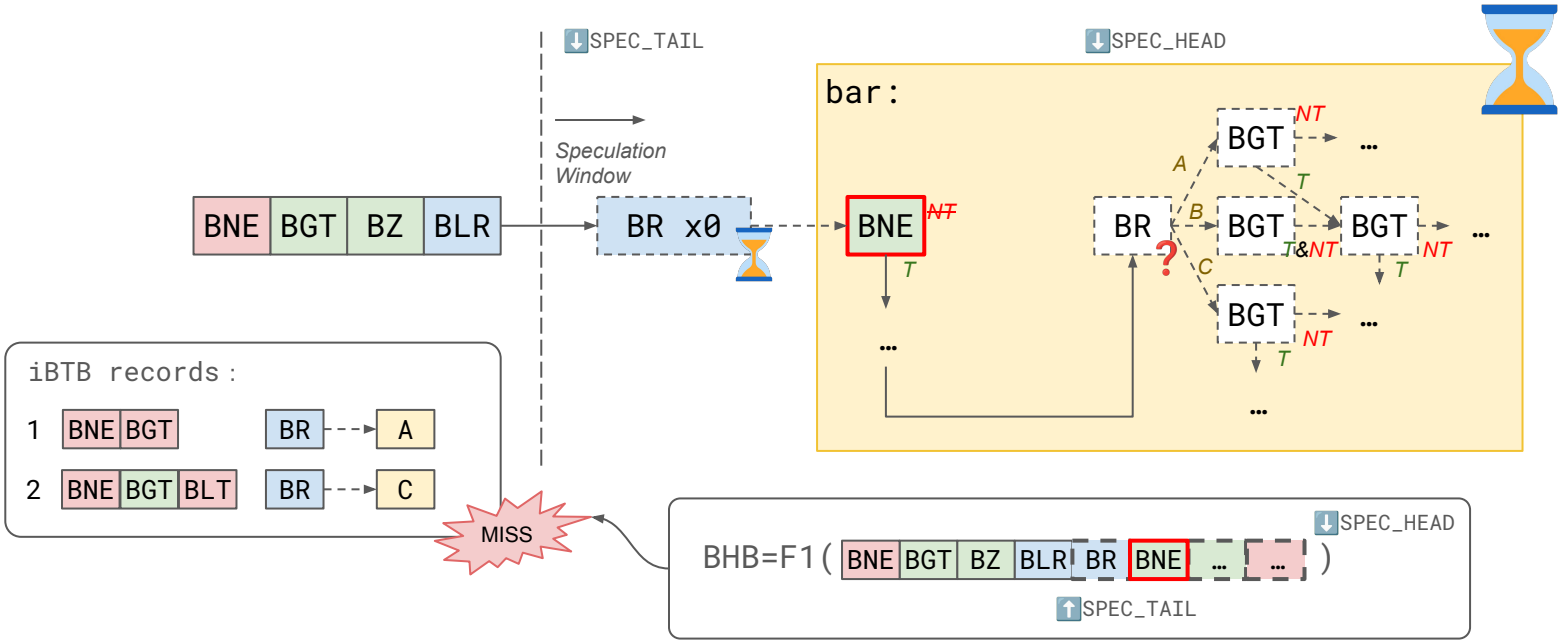
# Architectural footprint in speculated branch history

- BHB receives updates from architecturally committed branches and speculative predictions simultaneously.



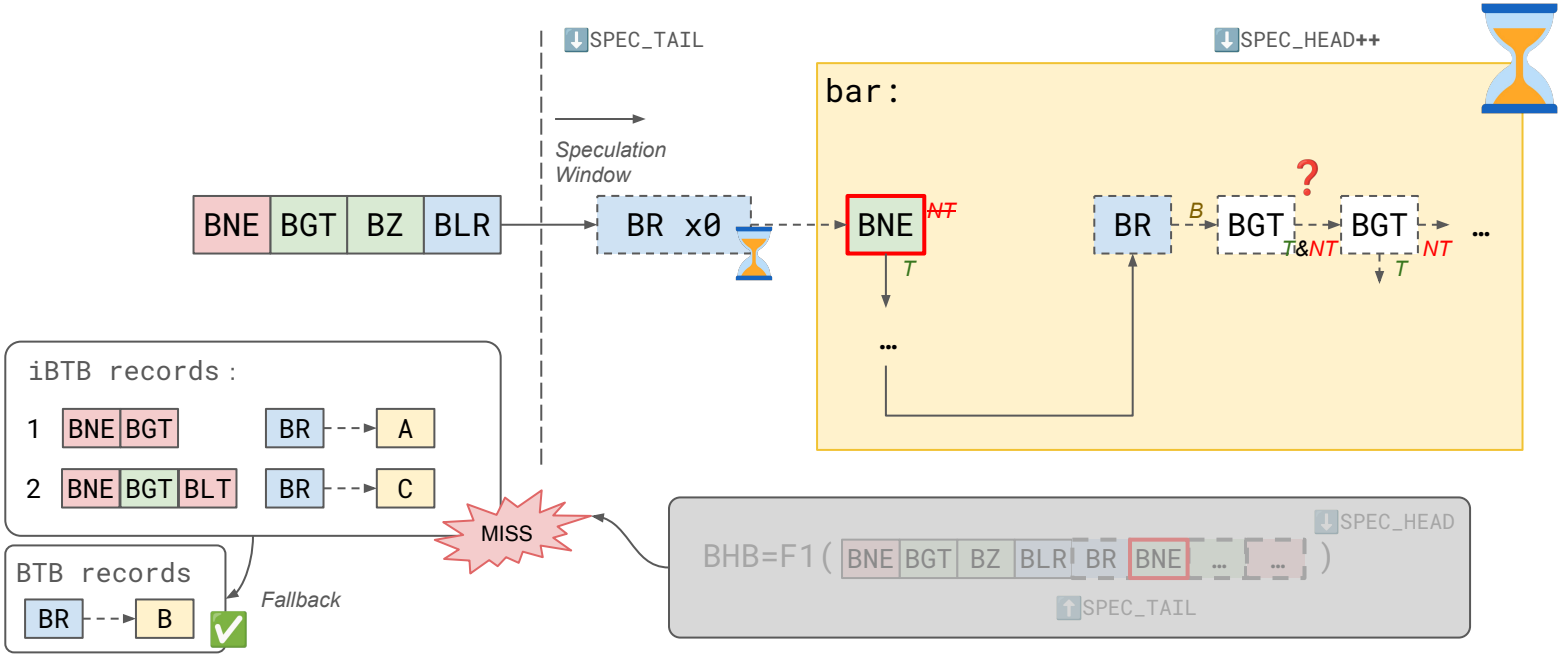
# Architectural footprint in speculated branch history

- Architecturally resolved outcomes may mismatch speculative predictions.

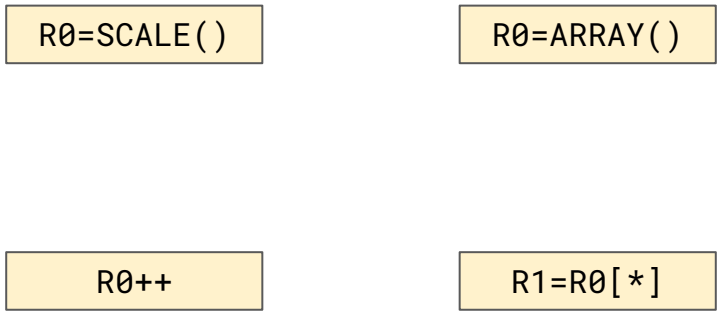


# Architectural footprint in speculated branch history

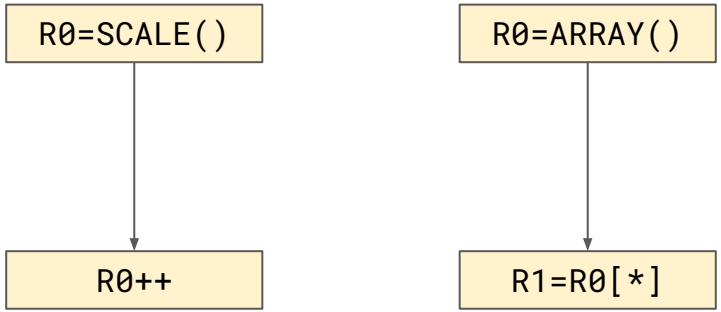
- Results in BTB miss, forcing fallback to PC-based predictions



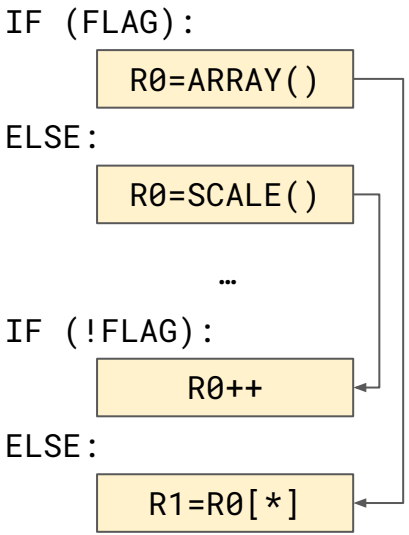
# eBPF program verification



# eBPF program verification



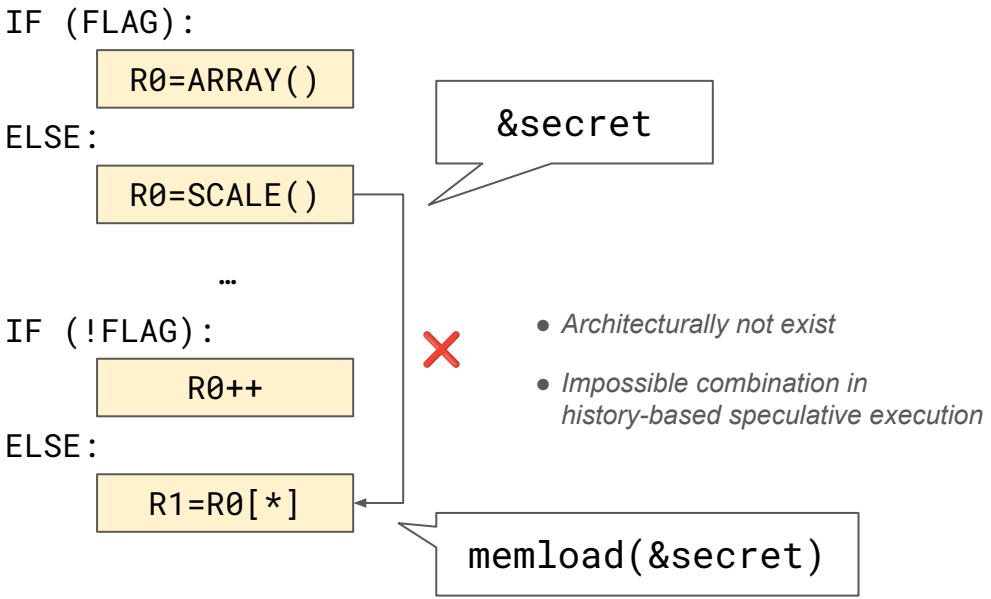
# eBPF program verification




✅

- *Accepted by eBPF verifier*
- *Recorded in BPU tables*

# eBPF program verification



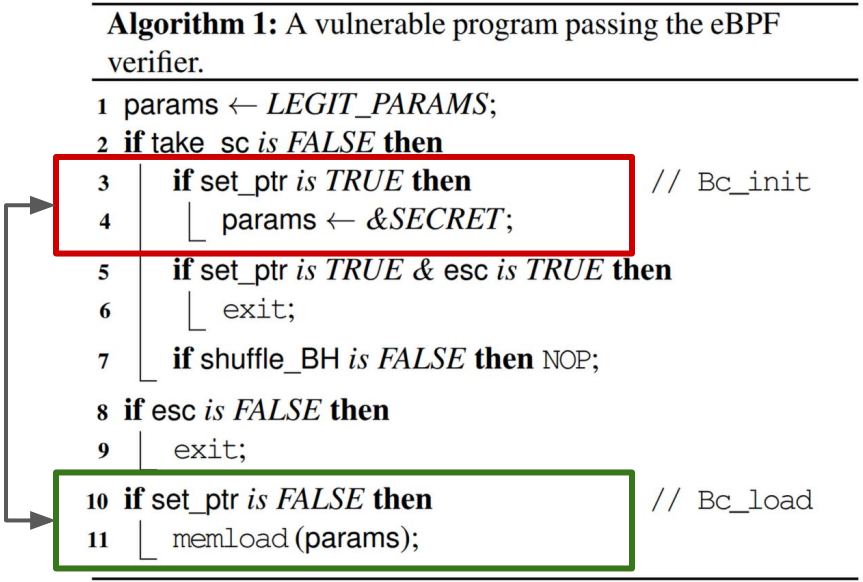
# eBPF program verification

```
IF (FLAG):
    R0=ARRAY()
ELSE:
    R0=SCALE()
    …
IF (!FLAG):
    R0++
ELSE:
    R1=R0[*]
```

- *Architecturally not exist*
- ~~*Impossible combination in history-based speculative execution*~~
- *Possible in PC-based speculative execution*

# Chimera eBPF Snippet

- Create mutually-exclusive blocks with complementary conditions;



**Algorithm 1:** A vulnerable program passing the eBPF verifier.

```
1  params ← LEGIT_PARAMS;
2  if take_sc is FALSE then
3      if set_ptr is TRUE then                    // Bc_init
4          params ← &SECRET;
5      if set_ptr is TRUE & esc is TRUE then
6          exit;
7      if shuffle_BH is FALSE then NOP;
8  if esc is FALSE then
9      exit;
10 if set_ptr is FALSE then                       // Bc_load
11     memload(params);
```

# Chimera eBPF Snippet

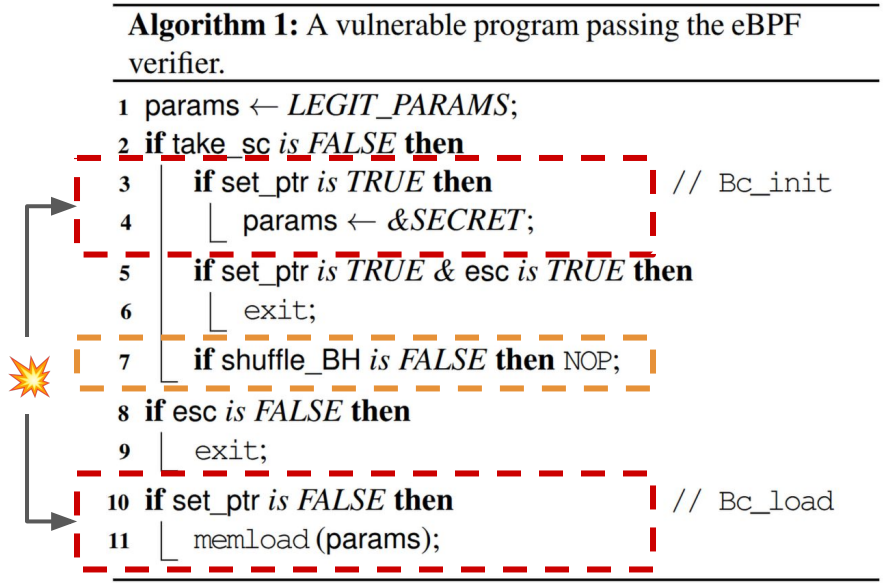
- Create mutually-exclusive blocks with complementary conditions;
- Insert BHB-shuffle branch to disrupt branch history;

**Algorithm 1:** A vulnerable program passing the eBPF verifier.

```
1  params ← LEGIT_PARAMS;
2  if take_sc is FALSE then
3      if set_ptr is TRUE then                    // Bc_init
4          params ← &SECRET;
5      if set_ptr is TRUE & esc is TRUE then
6          exit;
7      if shuffle_BH is FALSE then NOP;
8  if esc is FALSE then
9      exit;
10 if set_ptr is FALSE then                       // Bc_load
11     memload(params);
```

# Chimera eBPF Snippet

- Create mutually-exclusive blocks with complementary conditions;
- Insert BHB-shuffle branch to disrupt branch history;
- Force Line 10 to use PC-based fallback prediction;
- Execute incompatible blocks within same speculation window.



# Chimera eBPF Snippet

- Bypassed privilege-mode eBPF verifier and successfully loaded on platform;
- Successfully leaked kernel-space data on Cortex-A76/A78AE, Zen4, and Intel processors;
- Achieved 24,628 bit/s leakage rate on Cortex-A76.

**Algorithm 1:** A vulnerable program passing the eBPF verifier.

```
1  params ← LEGIT_PARAMS;
2  if take_sc is FALSE then
3      if set_ptr is TRUE then                    // Bc_init
4          params ← &SECRET;
5      if set_ptr is TRUE & esc is TRUE then
6          exit;
7      if shuffle_BH is FALSE then NOP;
8  if esc is FALSE then
9      exit;
10 if set_ptr is FALSE then                       // Bc_load
11     memload(params);
```

# Outline

1. Background
2. Bias-Free Branch Prediction (only Cortex-A72)
3. Branch History Speculation
4. Chimera Snippet and eBPF
5. **Conclusion**

# Conclusion

| | Bias-Free Branch Prediction | | Branch History Speculation | |
|---|---|---|---|---|
| | Spectre-BSE | BiasScope | Spectre-BHS | Chimera |
| Arm A72 | ✅ | ✅ | ✅ | |
| Arm A76/78AE | | | ✅ | ✅ |
| AMD Zen4 | | | ✅ | ✅ |
| Intel Gen10+ P/E | | | ✅ | ✅ |

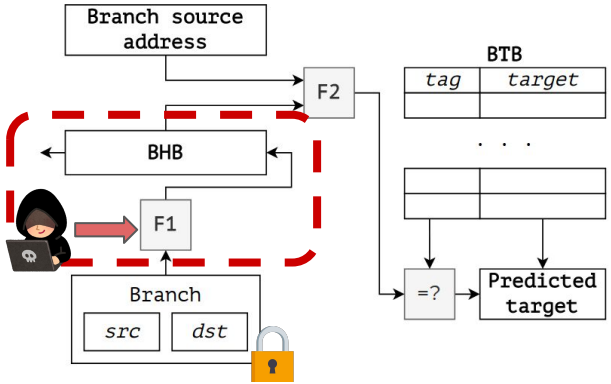



Figure 1: A simplified diagram of indirect branch prediction logic.

- Simply isolating or sanitizing the BHB is not fully effective in fixing this problem.
- Incomplete mitigations leave room for influencing how the BHB is updated.


# Thanks!

Artifact @ Github

{yuhui.zhu,alessandro.biondi}@santannapisa.it

# Exploiting Inaccurate Branch History in Side-Channel Attacks

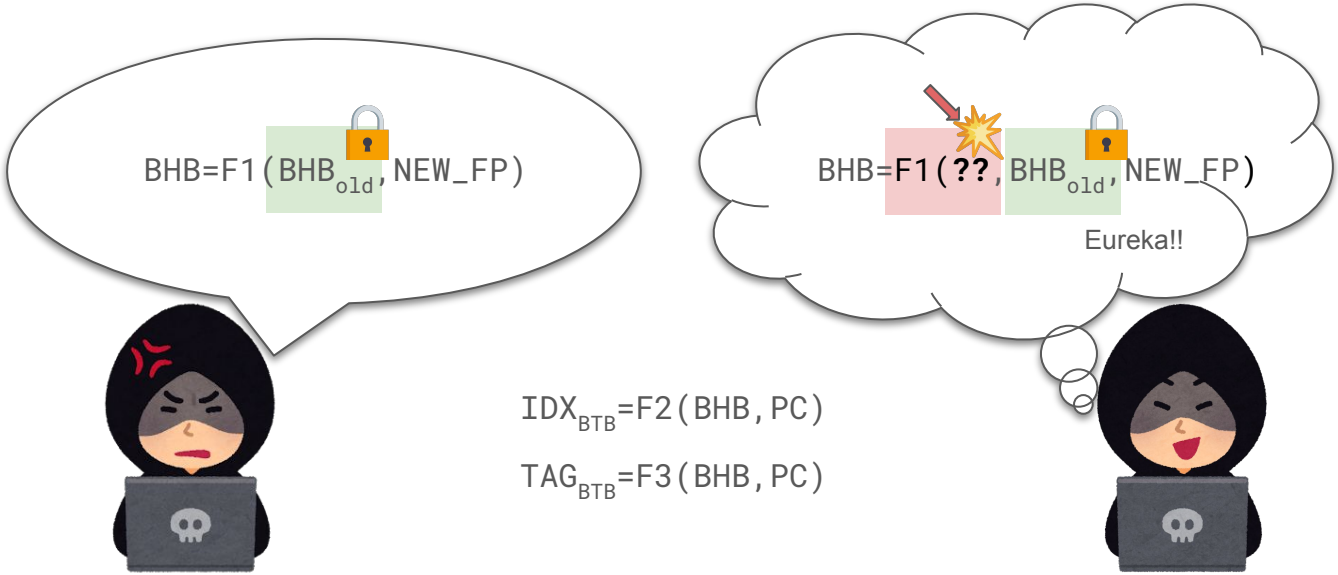
**Yuhui Zhu**[1,2], Alessandro Biondi[1]

[1]Scuola Superiore Sant'Anna, [2]Scuola IMT Alti Studi Lucca

Sant'Anna
School of Advanced Studies – Pisa

# TL;DR

- Simply isolating or sanitizing the *Branch History Buffer* (BHB) is not fully effective in fixing *Branch History Injection* (BHI) (Barberis2022).




(Barberis2022) E. Barberis, P. Frigo, M. Muench, H. Bos, and C. Giuffrida, “Branch history injection: On the effectiveness of hardware mitigations against Cross-Privilege spectre-v2 attacks,” in 31st USENIX security symposium (USENIX security 22), Boston, MA: USENIX Association, Aug. 2022, pp. 971–988.

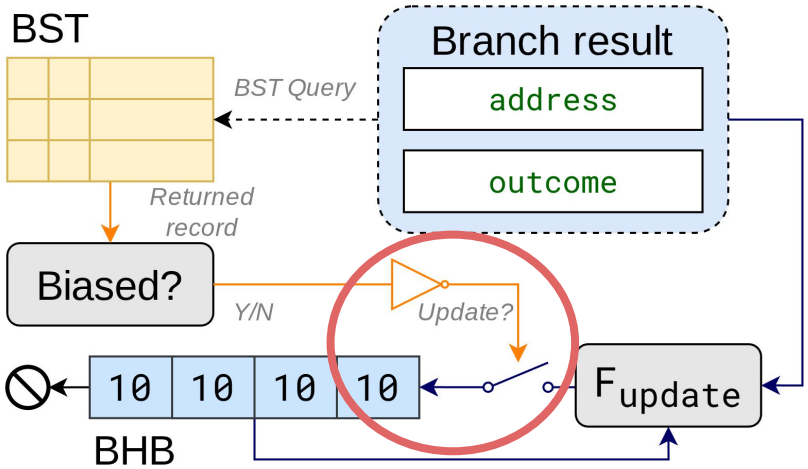
# TL;DR

- This work introduces four new attacks based on producing *inaccurate branch history*: Spectre-BSE, BiasScope, Spectre-BHS, and Chimera

| | *Bias-Free Branch Prediction* | | *Branch History Speculation* | |
|---|---|---|---|---|
| **Tested CPU** | **Spectre-BSE** | **BiasScope** | **Spectre-BHS** | **Chimera** |
| Arm A72 | ✅ | ✅ | ✅ | |
| Arm A76/78AE | | | ✅ | ✅ |
| AMD Zen4 | | | ✅ | ✅ |
| Intel Gen10+ P/E | | | ✅ | ✅ |

# Bias-Free Branch Prediction & BHB (Cortex-A72)

- Goal: Save BHB space by only recording *informative* indirect branches.
- Filter criterion: record branches with at least 2 targets (non-biased branch).
- *Branch Status Table* (BST): a cache to keep track of non-biased branches.



# Bias-Free Branch Prediction & BHB (Cortex-A72)

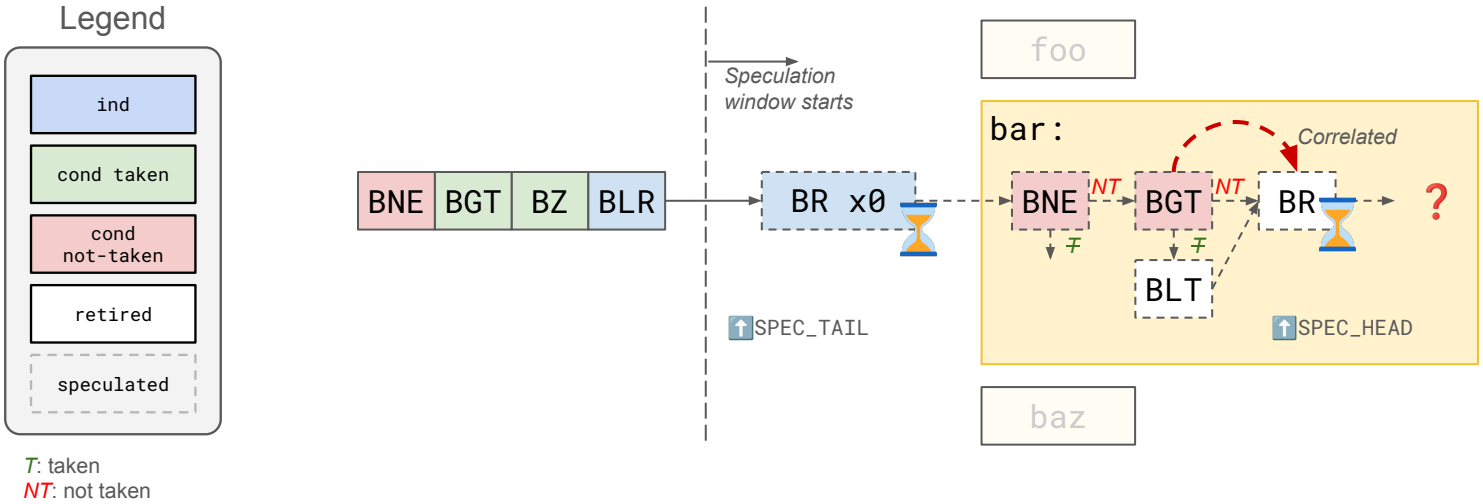
We exploited BST evictions in two directions:

- **Spectre-BSE (Branch Status Eviction) [CVE-2024-10929 by Arm*]**
  Sub-variant of BHI, indirectly controls the generated BHB value by manipulating the presence of BST entries.
- **BiasScope**
  Cross-context control flow side channel, exploits conflicts and evictions in the BST entries to infer branch behavior.

* Arm has also reported that the Cortex-A73 and Cortex-A75 are affected by this vulnerability.

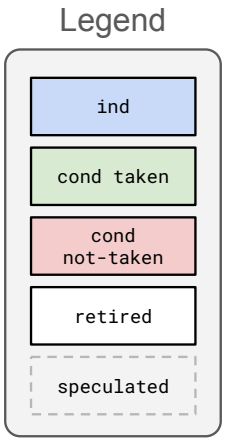
# Branch History Speculation (BHS)

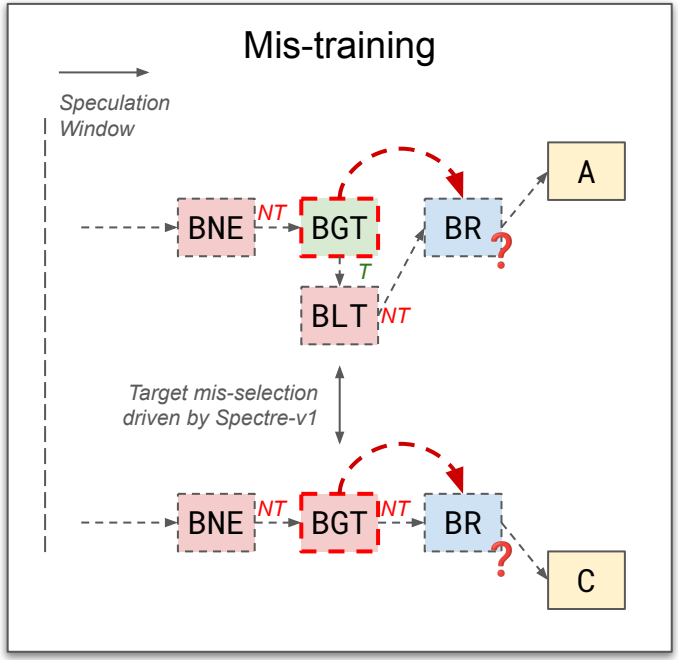
- BHB updated *using speculated branch outcomes* within speculation windows.
  - Support deeper and cascaded speculative execution;
  - Speculated branch outcomes should inform future predictions to maintain accuracy.

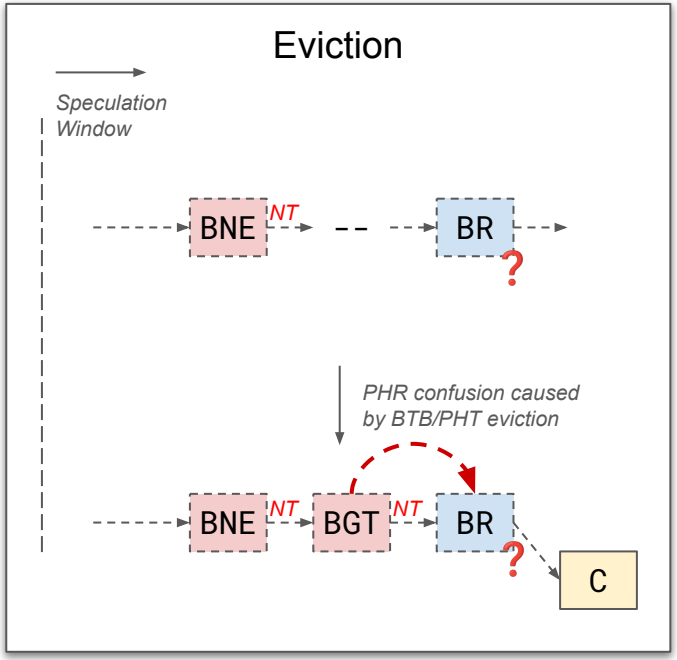


# Spectre-BHS

- **Attack:** Craft a distorted BHB value by evicting or mis-training Branch Prediction Unit (BPU) records inside the speculation window.

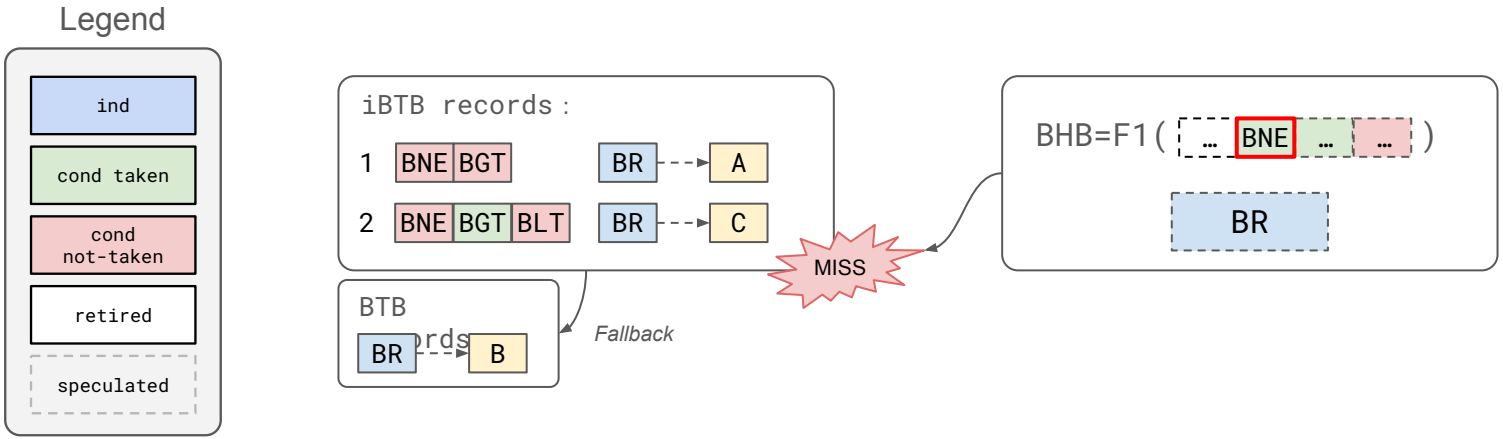


# Chimera: Construct a Non-Exist Speculation Path

- Mixed source of BHB updates under BHS scheme:
  - Speculated results;
  - Architecturally confirmed results.
- *Architecturally confirmed results* may “shuffle” the BHB:
  - Generate a pattern that is never recorded by the BPU;
  - Force the BPU to ignore correlation of branches;
  - Fallback to PC-based prediction and allow a speculation path that *never architecturally existed*.



# Chimera: Construct a Non-Exist Speculation Path

- **Chimera**: Bypass the rigorous *compile-time* control-flow checks in eBPF.
  - Execute incompatible blocks within the same speculation window to create an architecturally non-existent execution path.

**Algorithm 1:** A vulnerable program passing the eBPF verifier.

```
1  params ← LEGIT_PARAMS;
2  if take_sc is FALSE then
3      if set_ptr is TRUE then                    // Bc_init
4          params ← &SECRET;
5      if set_ptr is TRUE & esc is TRUE then
6          exit;
7      if shuffle_BH is FALSE then NOP;
8  if esc is FALSE then
9      exit;
10 if set_ptr is FALSE then                       // Bc_load
11     memload(params);
```


# Thanks!

Artifact @ Github

{yuhui.zhu,alessandro.biondi}@santannapisa.it