2024
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
EUROPE
MUNICH, GERMANY
OCTOBER 15-16, 2024

A Detailed Tour of IEEE standard P3164

Miltos Grammatikakis, Hellenic Mediterranean University

Speaker: Jörg Bormann, Siemens EDA

accellera
SYSTEMS INITIATIVE

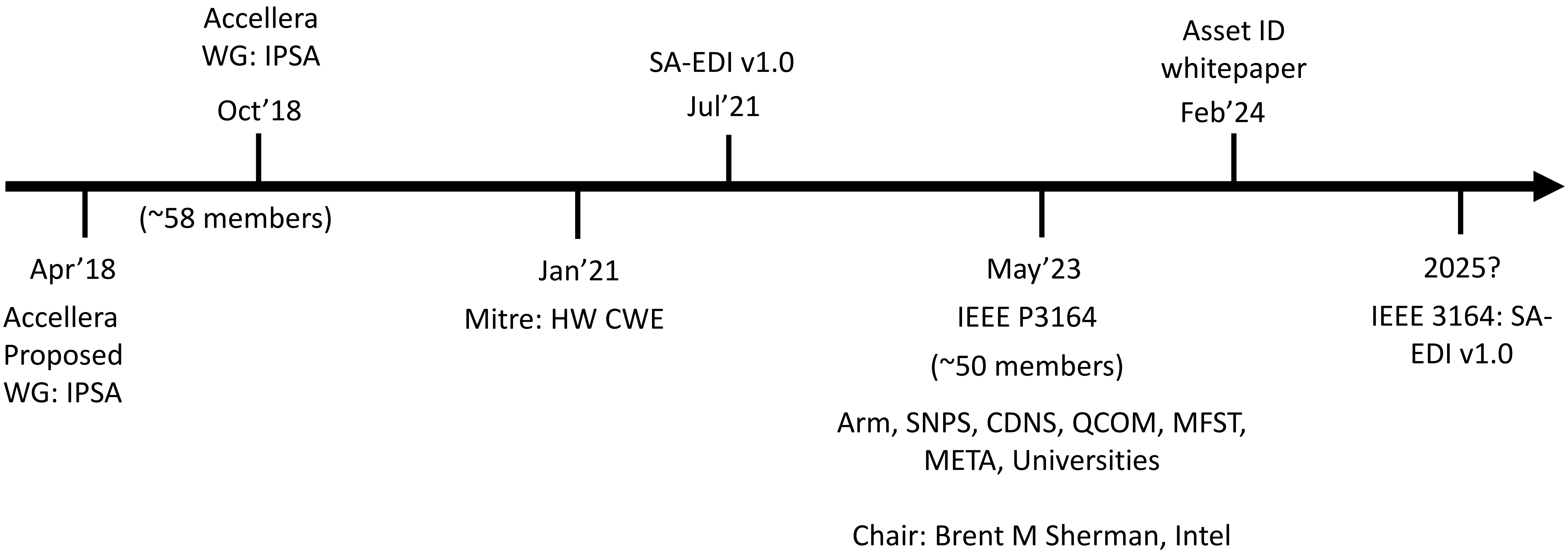
Agenda

- Problem Statement 1: Security Assurance collaterals of an IP
- Introduction to IEEE P3164 and SA-EDI standard
- Problem Statement 2: Asset Identification
- Asset Identification Whitepaper

Problem Statement (Part I)

- No mechanism of binding security assurance (SA) collateral to an IP
  - Missing verification of SA
- Unable to perform any data mining (e.g., common threats, security objectives, etc.)
- Lacking consistent quality in SA collateral
  - Different IP providers produce different collateral/formats (e.g., doc, ppt, pdf, xls, etc.)

Timeline



Apr’18

Accellera Proposed WG: IPSA

Accellera WG: IPSA

Oct’18

(~58 members)

Jan’21

Mitre: HW CWE

SA-EDI v1.0

Jul’21

May’23

IEEE P3164

(~50 members)

Arm, SNPS, CDNS, QCOM, MFST, META, Universities

Chair: Brent M Sherman, Intel

Asset ID whitepaper

Feb’24

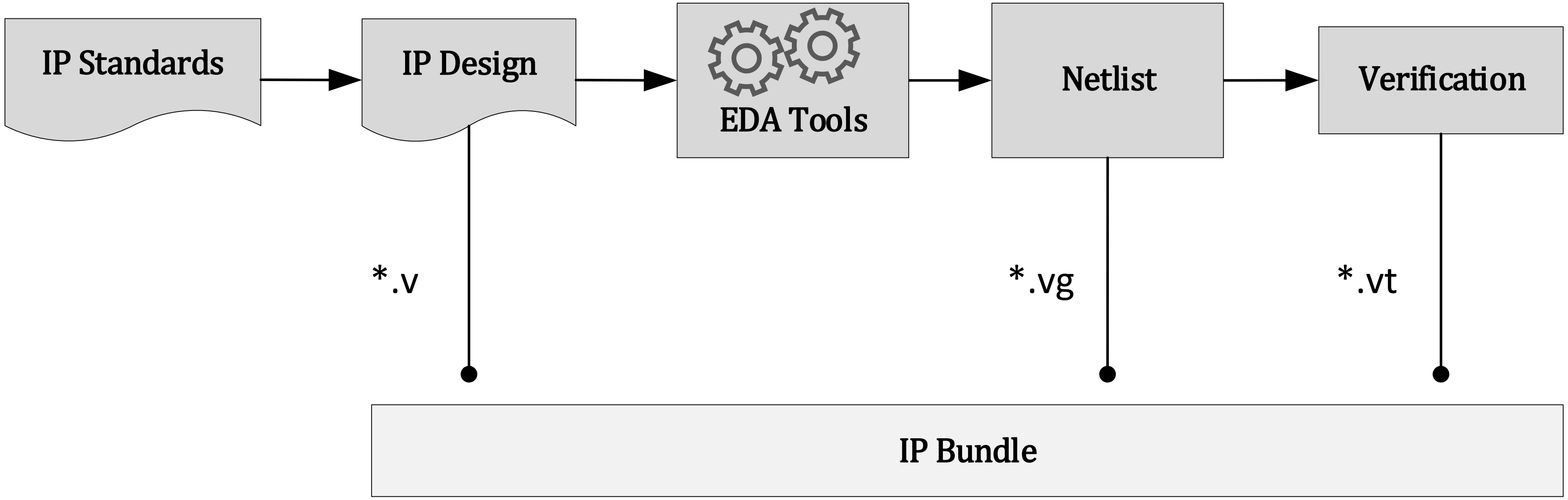
2025?

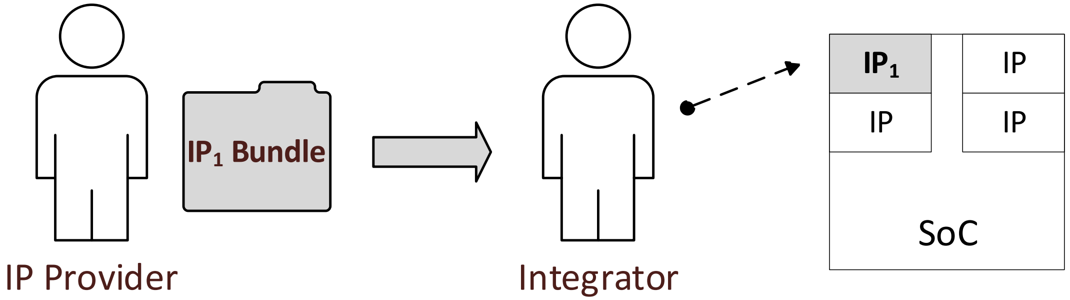
IEEE 3164: SA-EDI v1.0

SA-EDI Standard

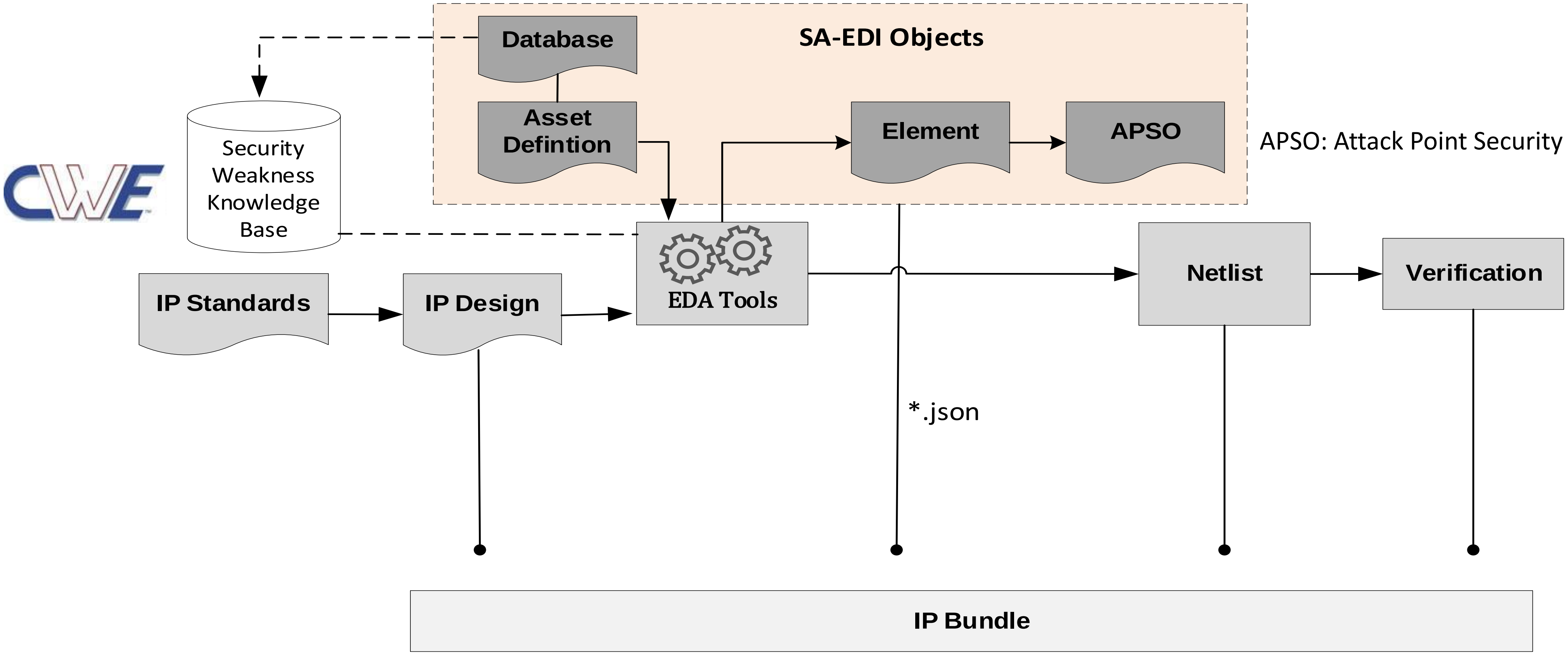
- Released July’21: https://www.accellera.org/
  - 21 authors, 11 companies
- Properties:
  - Uses JSON data modeling
    - Required fields helps consistency
    - Expansion supported for proprietary information
  - Binds the data objects to the RTL
    - Automatable and verifiable
  - Outside the design so can be applied to existing IP
  - Low overhead: Only 4 data object types

Today’s IP Design Flow





SA-EDI Data Objects

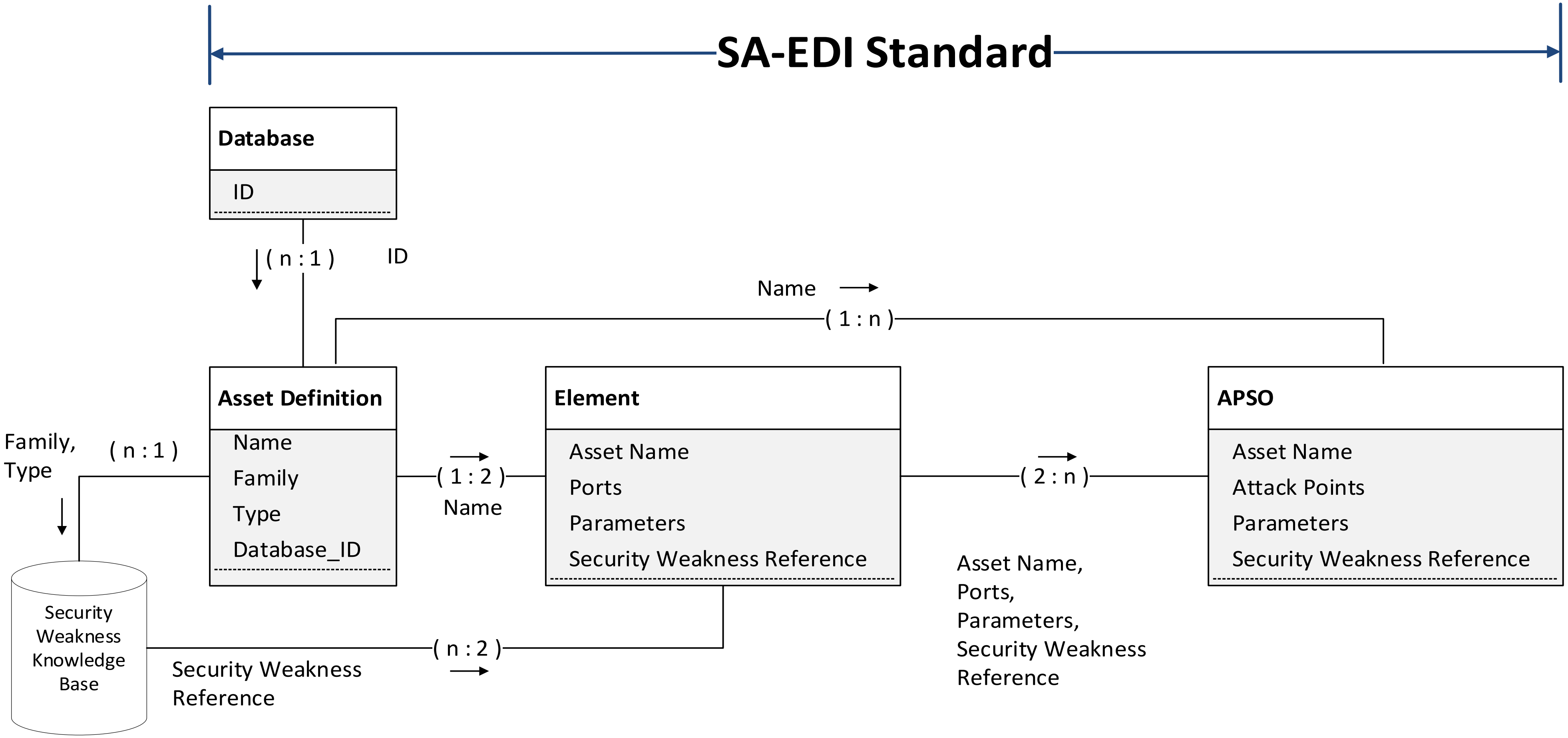


APSO: Attack Point Security Objective

SA-EDI Data Objects

- Database (Manual)
  - Key attributes defining a security weakness database (e.g., CWE)
- Asset Definition (Manual)
  - Root object that identifies assets in the IP
    - Asset = anything of value/importance (e.g., security objective)
- Element (Automated)
  - Identifies input/output ports and configuration parameters that can influence and/or observe the asset
- APSO - Attack Points Security Objective (Manual)
  - Assigns a security objective and attack points (Elements) to an asset
    - Confidentiality, Integrity, Availability
  - Identifies conditions that might violate the security objective

Data Object Associations



SA-EDI Standard

Database

ID

(n : 1)

ID

Name

(1 : n)

Asset Definition

Name

Family

Type

Database_ID

Family, Type

(n : 1)

(1 : 2)

Name

Element

Asset Name

Ports

Parameters

Security Weakness Reference

(2 : n)

APSO

Asset Name

Attack Points

Parameters

Security Weakness Reference

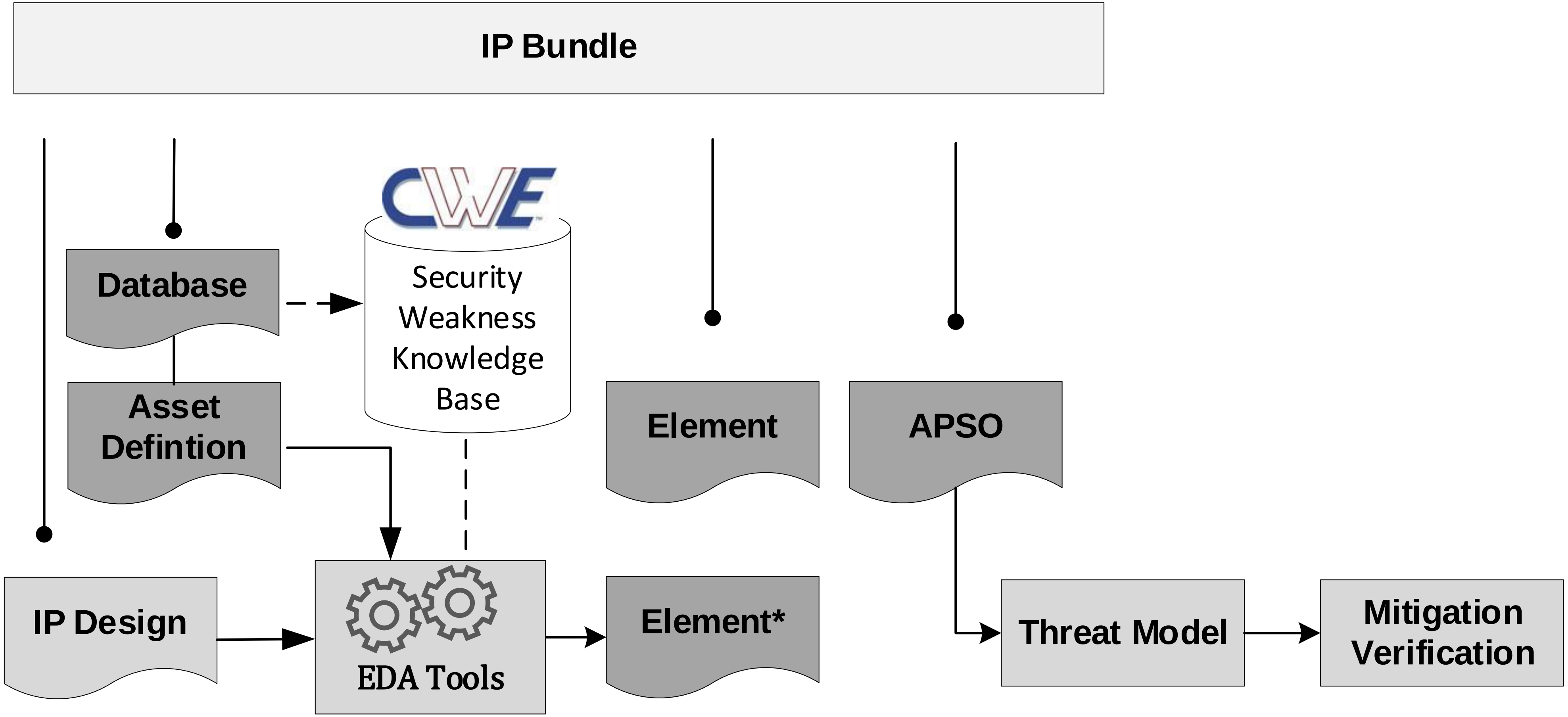
Asset Name, Ports, Parameters, Security Weakness Reference

Security Weakness Knowledge Base

(n : 2)

Security Weakness Reference

IP Bundle: Integrator

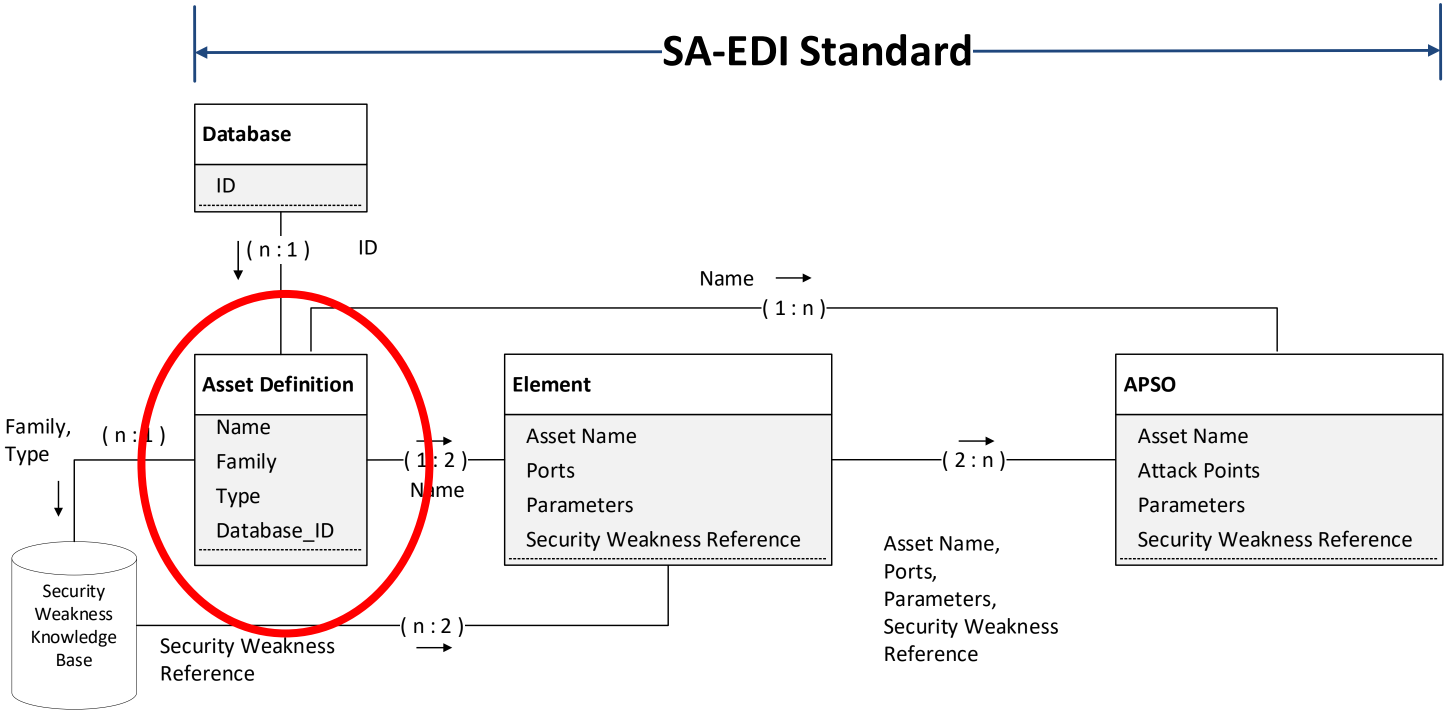


IP Integrator: Verifies; Threat Model

1. Verify the existence of the SA data objects
2. Using RTL source and Asset Definition object, re-generates the Element object(s) (E*)
3. Compares E* with the Element object(s) in the IP Bundle (E)
   - E* == E then SA collateral matches RTL. Use IP.
   - E* != E then stop; RTL/SA collateral out of sync.
4. Decide which APSO objects are in scope of the SoC/IC
   - Becomes part of the product’s threat model
5. Identify additional APSO objects for integration
   - Becomes part of the product’s threat model
6. Performs verification on the threat model

Problem Statement (Part II)

- What makes something an asset and how do I identify assets in my IP?
- Asset: Anything of value or importance that is used, produced, or protected within the IP



Comments

- “Everything in my IP is important therefore everything is an asset”
  - Creates 100 Asset Definition objects which yields 200 Element objects which yields 400 APSO objects. Humans can not consume 400 JSON objects to create a threat model for their IC.

- “My IP makes no security claims therefore there aren’t any assets”
  - Information coming into and/or existing the IP could require a security objective once integrated. IP owner needs to assume the IC may have security requirements.

IEEE P3164: Asset Identification Whitepaper

- Two methodologies:
  - Conceptual and Structural Analysis (CSA)
    - Using conceptual (high-level) assets to identify structural assets in the RTL
  - Points of Influence and Observation (PIO)
    - Using conceptual (high-level) assets and points of observation/influence to identify structural assets in the RTL

- Disclaimers
  - Methodologies are not mutually exclusive
  - Not the only methodologies for identifying assets
  - Both are subjective and not absolute

CSA Introduction

- Conceptual Asset: a high-level asset associated to the use-case flows of an IP which involves a security objective (CIA)
  - Ex: encryption key requires confidentiality
- Structural Asset: RTL material that physical supports a conceptual asset.
  - Ex: register, module, signal, etc.
- Asset Definition object: created based on the structural assets

Security Objectives: Chicken and the Egg

- IP developers typically do not know the security objectives of an IC/SoC. IPs are developed to a specification, not a use-case.
  - Ex: Same USB controller may be integrated into a phone, server, and military laptop. All three platforms have different security objectives but it is the same USB IP.

- Question: How to identify conceptual assets without knowing the security objectives?

- Answer: Assume the security objectives

Conceptual Assets: Questions

- Answer the following questions:

1. Assume the IP is to be integrated into an IC where confidentiality protections are required. Are there any elements in the IP that can leak or expose material that an Integrator may deem as confidential?
    - Is there any information, either as input or internally generated, that may be considered secret?

2. Assume the IP is to be integrated into an IC where integrity protections are required. Are there any elements in the IP that can modify material an Integrator may deem as sensitive?
    - Are there any state or configuration settings that need to be immutable during certain operations or modes?

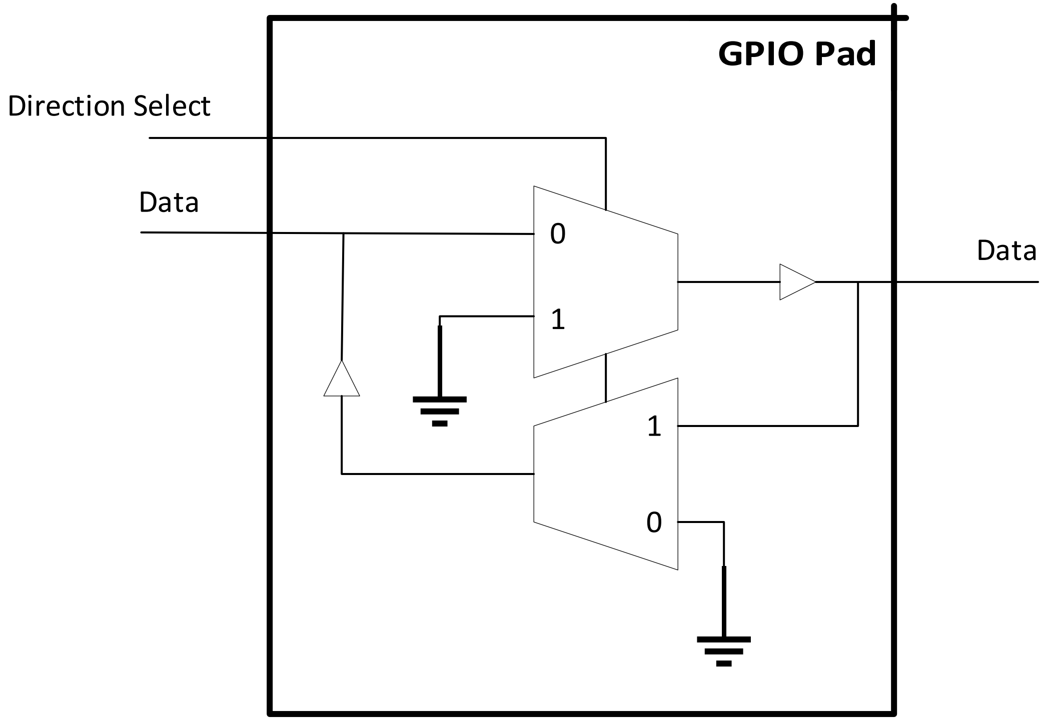
Conceptual Assets: Questions

- Answer the following questions:

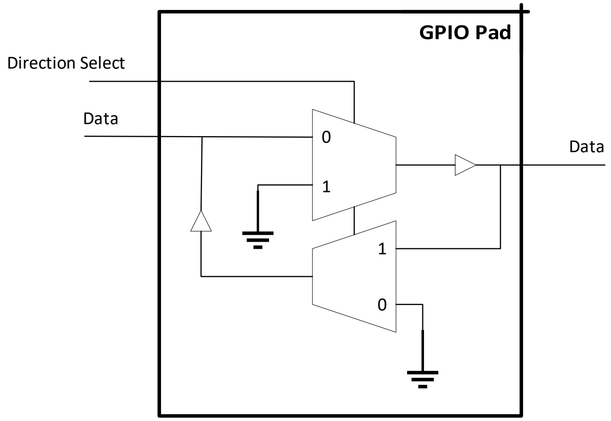
3. Are there any elements in the IP that if unavailable, would prohibit the operational behavior of the IP or IC?
   - Are there any elements that could gate an output port or the use of an input port? The focus should be on elements that may be impacted by a denial-of-service attack at the integration level.

4. Are there elements that could be impacted by behaviors at the integration level to undermine the functionality of the IP under normal operation?
   - For example, are there any privileged modes, overrides, bypass, test packet injection, etc. that can make the IP produce incorrect output? The focus should be on elements that may be compromised.

CSA Example: GPIO Pad

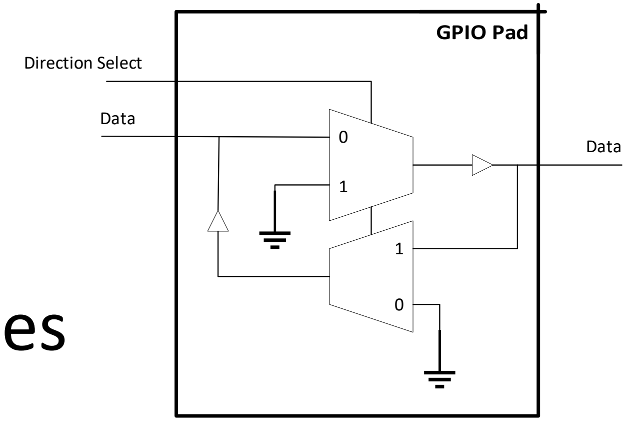


CSA Example: GPIO Pad



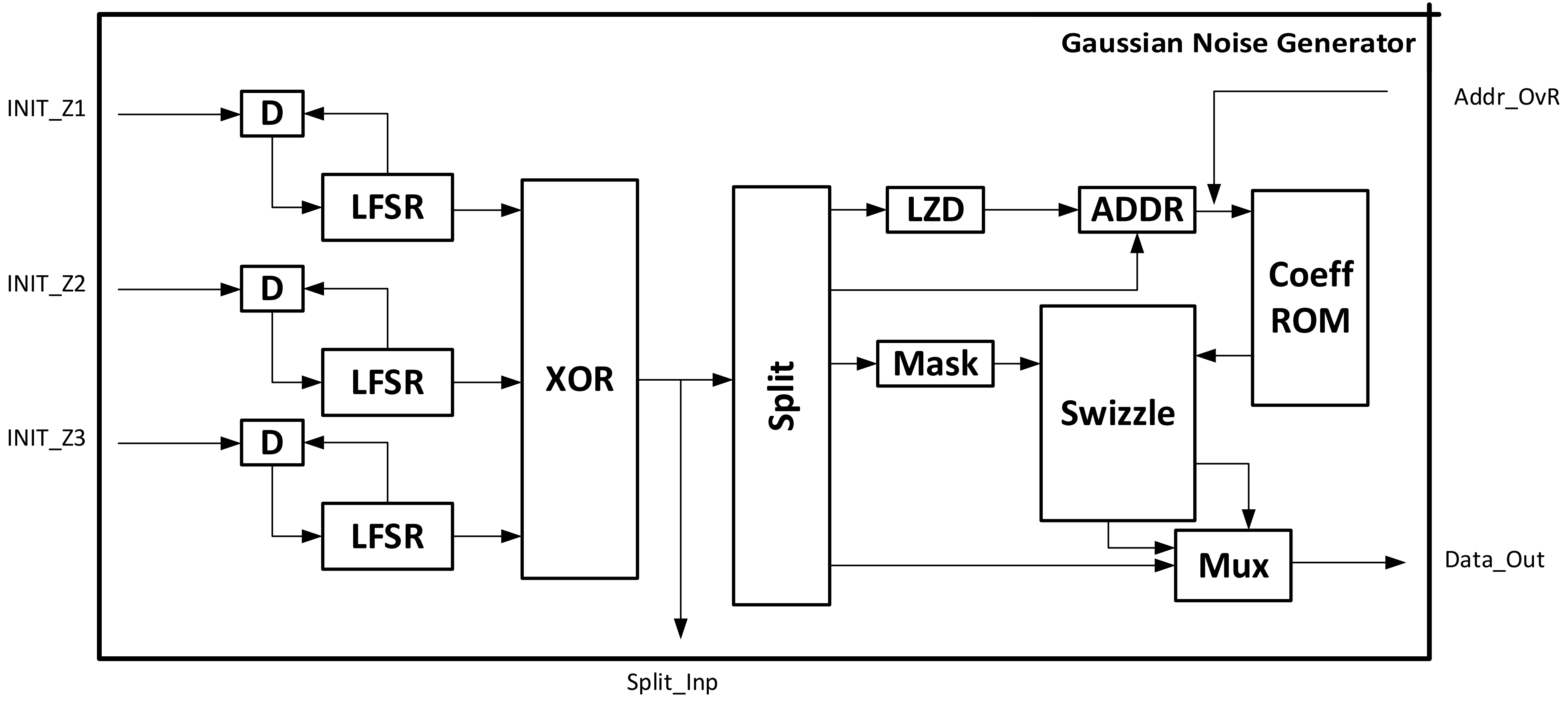
1. ***Confidentiality***: Are there any elements in the IP that can leak or expose material that may need confidentiality?
   - ❖ No. The gates inside the IP do not leak any information.
2. ***Integrity***: Are there any elements in the IP that can modify material an Integrator may deem as sensitive?
   - ❖ Yes. If the "Direction Select" was toggled during a runtime sample, the "Data" could also toggle in value, potentially producing an error. The mux gates can be considered as conceptual assets.
3. ***Availability***: Are there any elements in the IP that if unavailable can prohibit operational behavior?
   - ❖ Yes. The "Direction Select" can reverse the data flow on "Data" ports, which can be a denial-of-service. The elements impacted by this attack are the mux gates and should be considered as conceptual assets.
4. ***Undermined expected behavior***: Are there elements that could be impacted by behaviors at the integration level to undermine the functionality of the IP under normal operation?
   - ❖ No. The IP has no privilege or bypassing mechanisms that will alter its normal behavior.

CSA Example: GPIO Pad



- Questions #2 & #3 triggered so therefore the mux gates are the conceptual assets
- Structural assets: the RTL constructing these gates
- The RTL assets create the Asset Definition objects

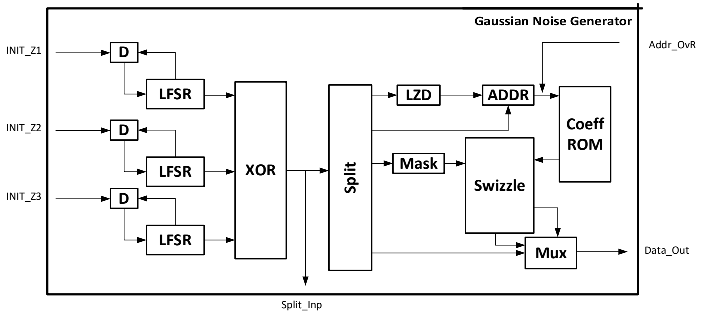
CSA Example: Gaussian Number Generator



LIU, GURANGZI. OPENCORES. (2014). GAUSSIAN NOISE GENERATOR. HTTPS://OPENCORES.ORG/PROJECTS/GNG

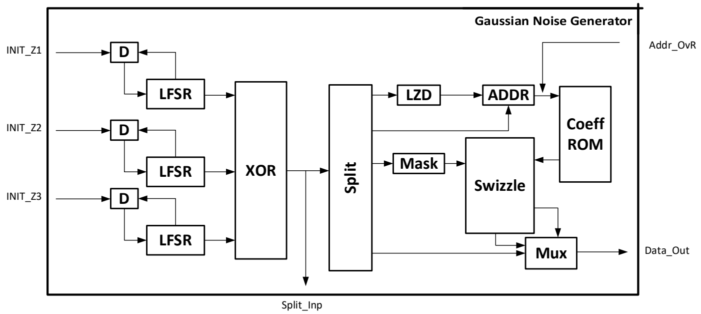
LZD: lead zero detector

CSA Example: Gaussian Number Generator



1. ***Confidentiality***: Are there any elements in the IP that can leak or expose material that may need confidentiality?
   - Yes. The output value of the XOR block can be deemed as a seed and if observed, may be used to predict the Gaussian Noise (GN). This assumes the IC considers GN as a secret. The conceptual asset would be the XOR and LFSR blocks.
2. ***Integrity***: Are there any elements in the IP that can modify material an Integrator may deem as sensitive?
   - Yes. The address into the Coeff ROM should not be modified once the “INIT_Z” inputs are set. Modifying the address to use a different coefficient than the one intended may reduce the randomness of the GN. Therefore, the conceptual assets would be the ADDR block and Coeff ROM.
3. ***Availability***: Are there any elements in the IP that can become unavailable, prohibiting operational behavior?
   - No. There isn’t a means at the integration level to disable or impede “Data_Out”.
4. ***Undermined expected behavior***: Are there elements that could be impacted by behaviors at the integration level to undermine the functionality of the IP under normal operation?
   - Yes. Input “Addr_OvR“ can force the IP to select an unintended coefficient which may produce an invalid GN on “Data_Out”, pending on the use-case. Therefore, Coeff ROM is a conceptual asset.

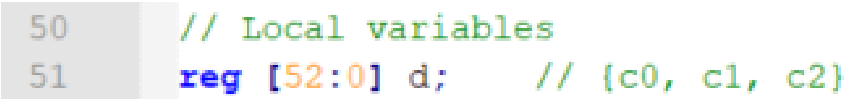
CSA Example: Gaussian Number Generator



Conceptual assets:

- XOR
- LFSR
- ADDR
- Coeff ROM

Structural example: output value of Coeff ROM is located in file gng_coef.v at line 51:

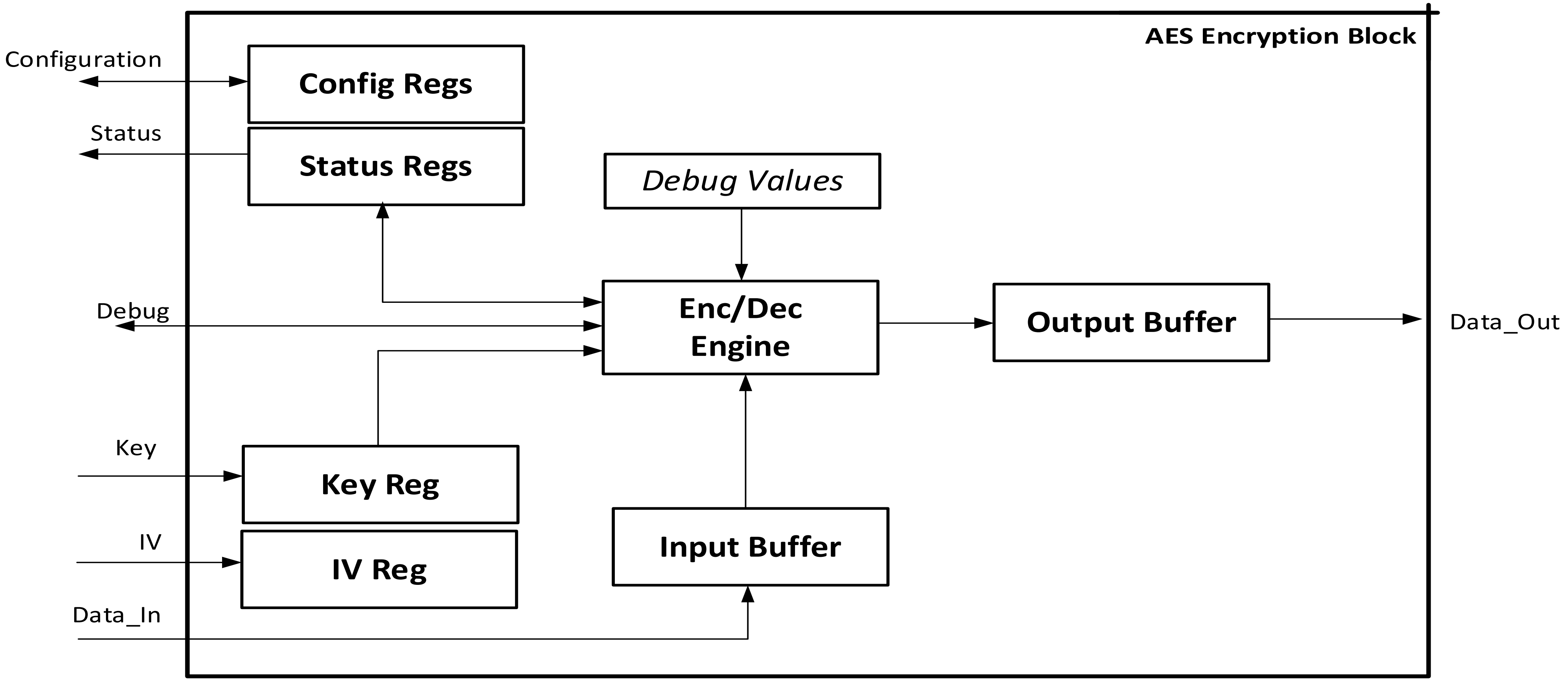


```
50   // Local variables
51   reg [52:0] d;    // {c0, c1, c2}
```

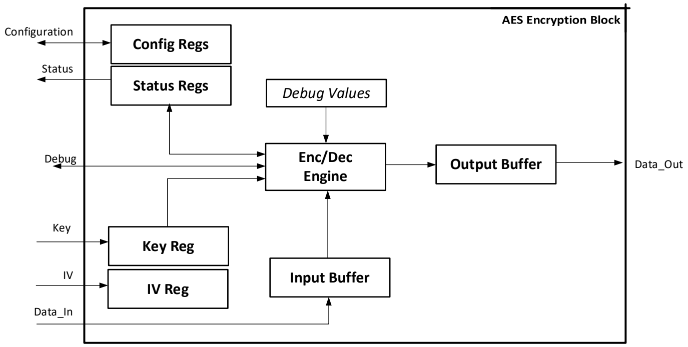
Asset Definition object:

```
{
 "Name" : "gng.gng_interp.gng_coef.d",
 "Description" : "Output from Coeff ROM",
 "Family" : ["Accelerator"],
 "Type" : ["Sensitive"],
 "Database_ID" : ["CWE VIEW: Hardware Design"]
}
```

CSA Example: AES Engine

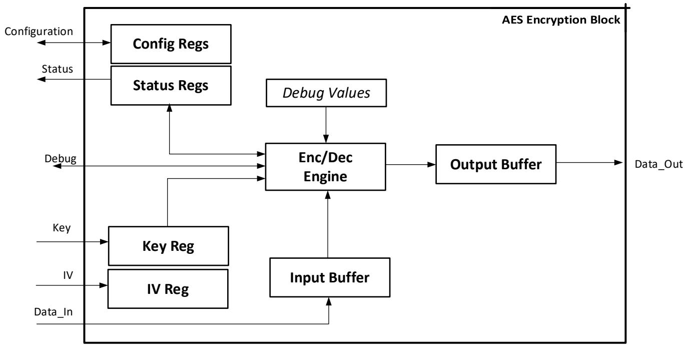


CSA Example: AES Engine



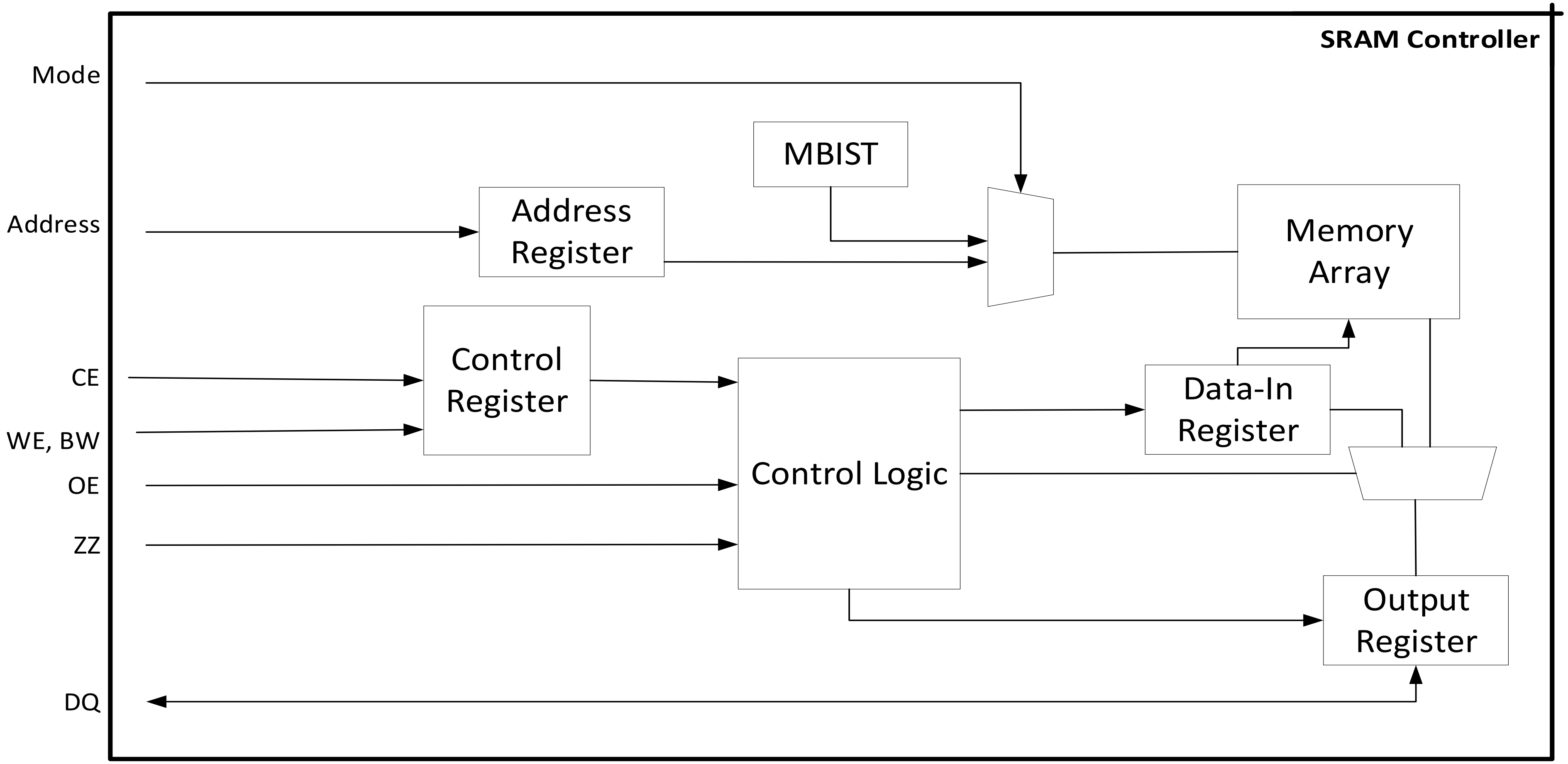
1. ***Confidentiality***: Are there any elements in the IP that can leak or expose material that may need confidentiality?
   1. Yes. Since this is a crypto IP, the plaintext data and key value are secrets. Therefore, any block in FIGURE 3 that supports these secrets will be a conceptual asset. These assets are Key Reg, Enc/Dec Engine, Input Buffer, and Output Buffer. Additionally, the Status Regs may leak confidential information since it provides information about the Enc/Dec Engine. Therefore, this block may also be considered a conceptual asset.
2. ***Integrity***: Are there any elements in the IP that can modify material an Integrator may deem as sensitive?
   1. Yes. When the Enc/Dec Engine is performing an operation, the key, IV, input data, and its configuration should not be modified. Therefore, Key Reg, IV Reg, Input Buffer, and Config Regs are conceptual assets that require integrity.
3. ***Availability***: Are there any elements in the IP that can become unavailable, prohibiting operational behavior?
   1. Yes. The debug interface allows complete control of the Enc/Dec Engine. Therefore, “Data_Out” can be blocked by this interface and thus making the Enc/Dec Engine a conceptual asset.
4. ***Undermined expected behavior***: Are there elements that could be impacted by behaviors at the integration level to undermine the functionality of the IP under normal operation?
   1. Yes. The debug interface allows the IP to encrypt/decrypt using the test key and IV values, which may result in a loss of security strength. Therefore, making the Enc/Dec Engine a conceptual asset.

CSA Example: AES Engine



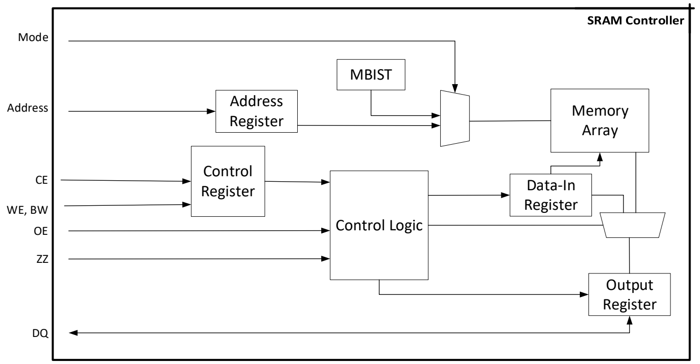
As identified in the questions above, every block in the IP, except the Debug Values block, can be considered as a conceptual asset. Therefore the RTL in these blocks would be the structural assets and require Asset Definition objects, which may be too numerous to comprehend. This is common for IPs that make security claims such as cryptography. One could assert that the entire IP is a structural asset whereas the top RTL module would be the Asset Definition object. This would reduce the Asset Definition objects into just one, which simplifies the analysis. Another approach, which is a modification to the CSA methodology and is detailed in section 4, may be to analyze the IP from an attack point perspective to identify assets. This approach could help identify assets that are false positives. Both approaches are acceptable and should result in the same APSO objects being created.

CSA Example: SRAM Controller



INTEGRATED SILICON SOLUTION, INC. OPENCORES. (2008). ZBT SRAM CONTROLLER. HTTPS://OPENCORES.ORG/PROJECTS/ZBT_SRAM_CONTROLLER

CSA Example: SRAM Controller



1. ***Confidentiality***: Are there any elements in the IP that can leak or expose material that may need confidentiality?

   ❖ Yes. If secret data is stored in the SRAM, then the Memory Array becomes an asset. Additionally, the Data-In Register and Output Register may also contain secret information that is readable at the integration level. The Address Register might contain information that may be considered a secret, pending on the use-case. Therefore, the conceptual assets are Memory Array, Data-In Register, Output Register, and Address Register.

2. ***Integrity***: Are there any elements in the IP that can modify material an Integrator may deem as sensitive?

   ❖ Yes. The Integrator may want a certain address range to be read-only. Therefore, this range will need integrity protections and thus making the Memory Array a conceptual asset.

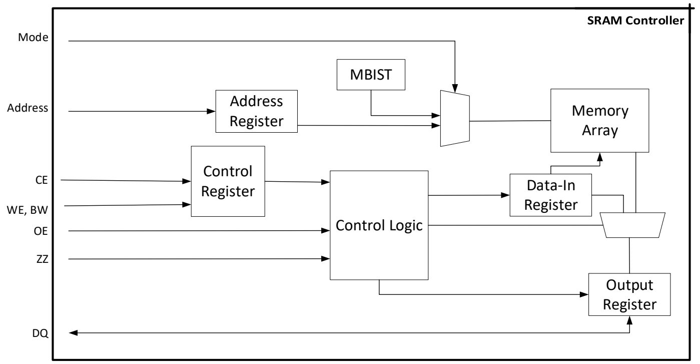
3. ***Availability***: Are there any elements in the IP that can become unavailable, prohibiting operational behavior?

   ❖ Yes. If MBIST is enabled, the Address signal is no longer an input into the Memory Array and thus preventing operational behavior. If the "ZZ" signal is asserted, the IP goes into sleep mode and prevents it from operation. Therefore, the Memory Array is a conceptual asset.

4. ***Undermined expected behavior***: Are there elements that could be impacted by behaviors at the integration level to undermine the functionality of the IP under normal operation?

   ❖ Yes. The MBIST operation makes the IP unusable while it is executing test patterns. Therefore, the Memory Array is a conceptual asset.

CSA Example: SRAM Controller



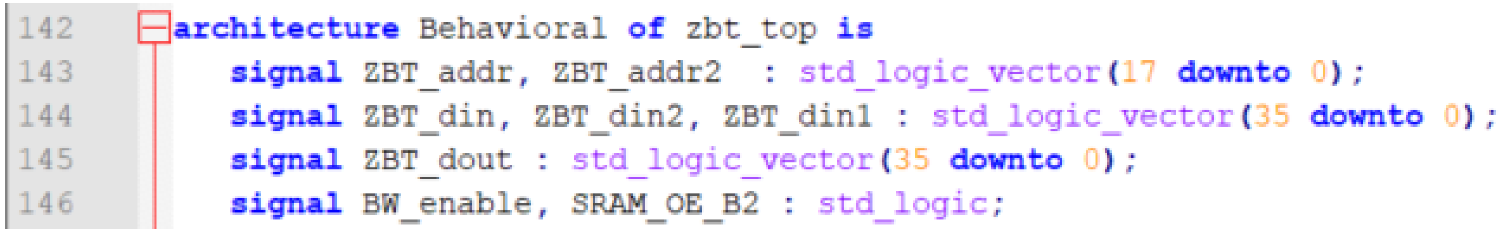
Conceptual assets:

- Memory Array
- Data-In Register
- Output Register
- Address Register

Asset Definition object:

```
{
  "Name" : "zbt_top.ZBT_addr",
  "Description" : "SRAM address requires confidentiality
protections",
  "Family" : ["Memories"],
  "Type" : ["Secret, Sensitive"],
  "Database_ID" : ["CWE VIEW: Hardware Design"]
}
```

Structural asset: address value in the Address Register is in file zbt_top.vhd at line 143:



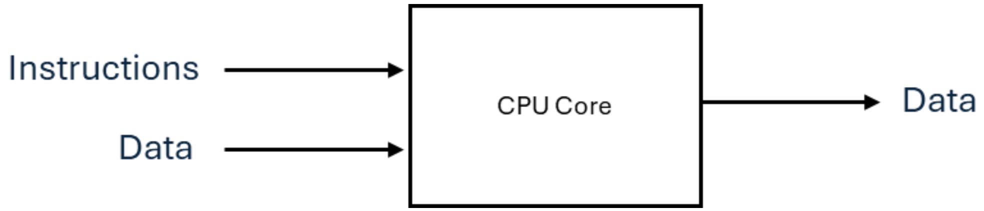
```
142  architecture Behavioral of zbt_top is
143      signal ZBT_addr, ZBT_addr2  : std_logic_vector(17 downto 0);
144      signal ZBT_din, ZBT_din2, ZBT_din1 : std_logic_vector(35 downto 0);
145      signal ZBT_dout : std_logic_vector(35 downto 0);
146      signal BW_enable, SRAM_OE_B2 : std_logic;
```

Alternative Approach

- CSA may be difficult to scale for complex IPs
  - Ex: CPU core
- CSA may produce false positives or too many structural assets for human comprehension
- Points of Influence and Observation (PIO) Methodology
  - Focus on inputs and outputs to identify conceptual assets
  - Walk through the IP where these conceptual assets can be influenced or observed to identify structural assets:
    1. Does the observation point expose any confidentiality of the conceptual asset?
    2. Does the influence point allow any modification of the conceptual asset?
    3. Can the observation and/or influence point prevent the conceptual asset from being available for functional operation?
    4. Does the observation and/or influence point have any special behaviors that can prevent the conceptual asset from being available for normal operation?

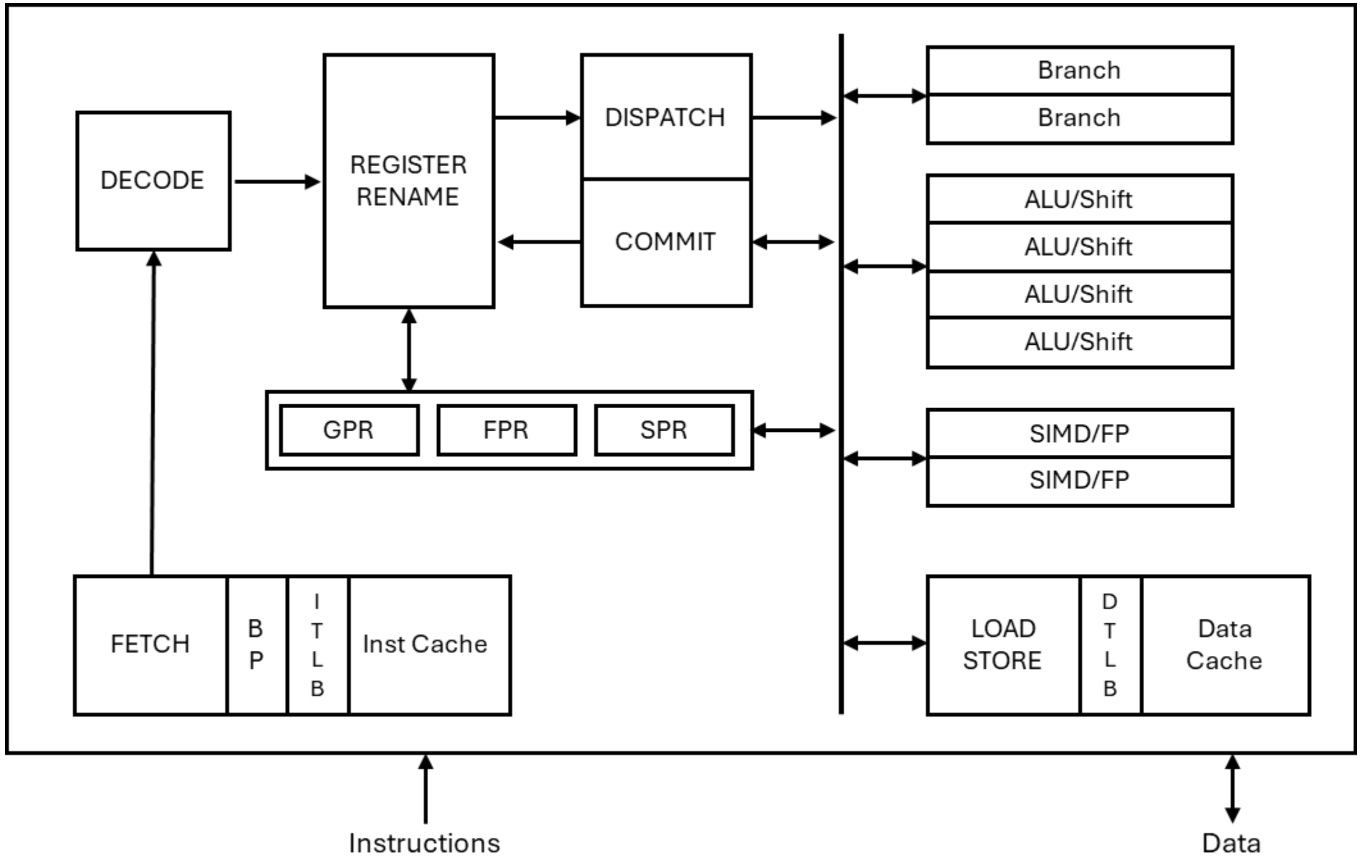
PIO Example: CPU Core

- Generic CPU core conceptual assets:
  - Instructions
  - Data (input and output)



PIO Example: CPU Core

- Registers:
  - General purpose
  - Floating point
  - Special
- Branch prediction
- Translation buffer
- Execution units
  - ALU
  - SIMD



PIO Example: CPU Core

- Walk through the detailed blocked diagram with focus on Instructions and Data to find the structural assets
- Ex: DCache

1. ***Confidentiality***: Does DCache expose any confidentiality of Data? – Yes. Caches are a shared resource that have been known to leak information under certain circumstances.
2. ***Integrity***: Does DCache allow any modification of Data? – No. The DCache by itself cannot modify data but it can replace when a store operation is requested. However, this is expected behavior and should not result in a “yes” to this question.
3. ***Availability***: Can DCache prevent Data from being available to functional operation? –Yes. Thrashing or exhausting the cache can prevent data from being available, in a timely fashion.
4. ***Undermined expected behavior***: Does DCache have any special behaviors that can prevent Data from being available for normal operation? – No. There are no features in the DCache that prevents data from being available.

PIO Example: CPU Core

- Triggered on questions #1 and #3, therefore structural assets exist in DCache
- Potential structural assets are the RTL logic for:
  - Replacement policy
  - DCache contents
  - Internal state

PIO Example: CPU Core

Conceptual Asset: Instructions			
Observation/Influence Pt	Rationale	Structural Asset(s)	Objective at Risk
ICache	Caches, if not protected, can be used as covert/side channels to exfiltrate data	ICache replacement policy, ICache contents, ICache internal state	Confidentiality, Availability
BP (Branch Predictor)	BP, if not protected, can be used to influence the flow of control. BP also allows speculative execution which opens a possibility to exploit transient execution attacks	Branch Prediction History and Target Addresses	Integrity
ITLB	Caches, if not protected, can be used as covert/side channels to exfiltrate data	Memory Mapping, ITLB contents, ITLB replacement policy	Confidentiality, Availability
Conceptual Asset: Data			
GPR, FPR	General purpose registers are typically shared between multiple processes	Registers	Confidentiality
Functional Units (ALU/Shift, Branch, SIMD, etc.)	The processing time can reveal the data processed if directly dependent on the data itself	Source and data registers	Confidentiality
DCache	Caches if not protected can be used as covert/side channels to exfiltrate data	DCache replacement policy, DCache contents, DCache internal state	Confidentiality, Availability
DTLB	Caches, if not protected, can be used as covert/side channels to exfiltrate data	Memory Mapping, DTLB contents, DTLB replacement policy	Confidentiality, Availability

Summary

- Two methods to help identify IP assets
  - Conceptual and structural analysis (CSA)
  - Points of Influence and Observation (PIO)

- They are not mutually exclusive
- Can be used with other methodologies
- No right or wrong approach

What’s Next?

- Automation
- Additional fields
- Expanded functionality
- Tools?

Thank You & Questions

- Information links:
  - Accellera IPSA workgroup: https://www.accellera.org/activities/working-groups/ip-security-assurance
  - IEEE P3164 workgroup: https://sagroups.ieee.org/3164/
  - Asset Identification Whitepaper: https://ieeexplore.ieee.org/document/10496567
  - SA-EDI: https://www.accellera.org/downloads/standards/ip-security-assurance
  - CWE: https://cwe.mitre.org/