SARA ACHOUR / MIT CSAIL
MARTIN RINARD / MIT CSAIL

PROGRAMMING ANALOG DEVICES WITH JAUNT AND ARCO

Dynamical Systems

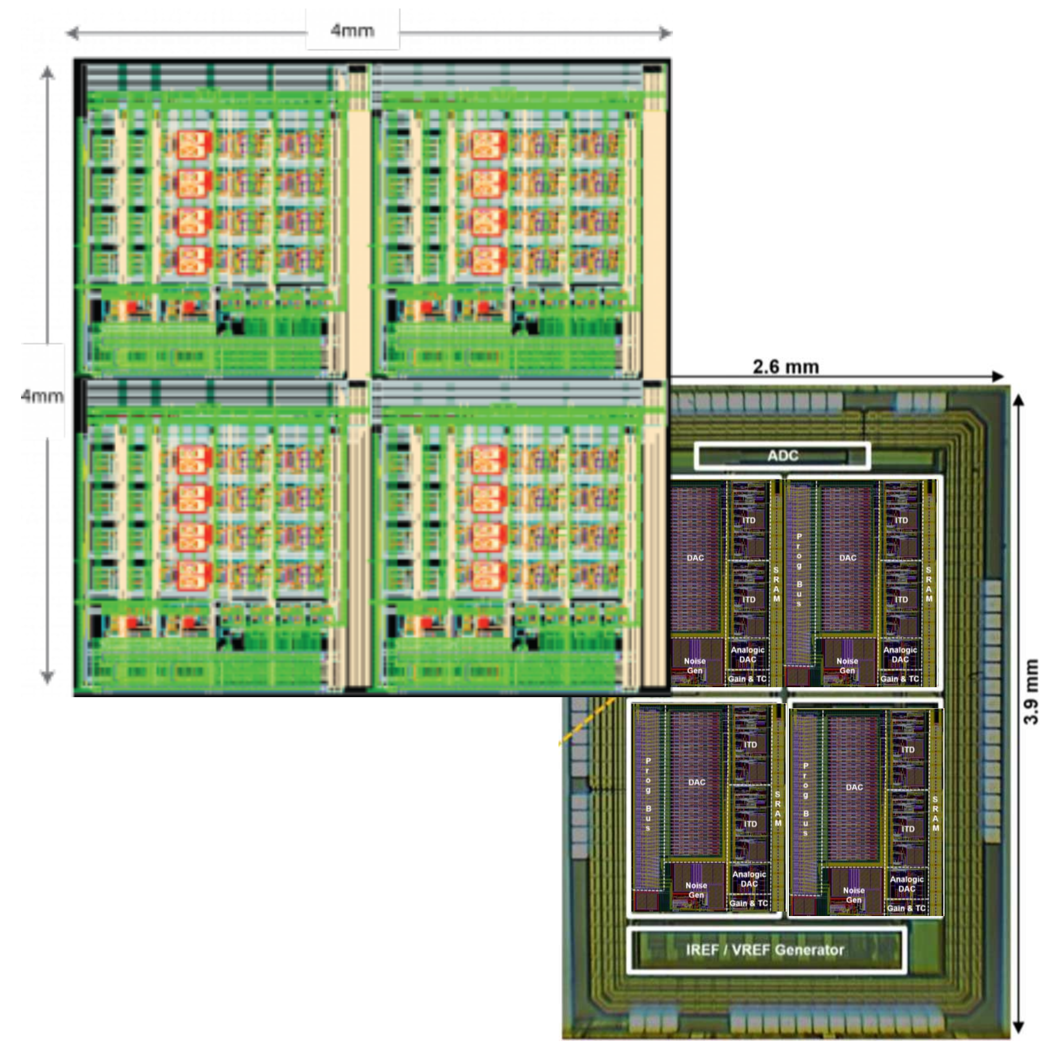
$$x'' + c_1 x' + c_2 x'' = 0$$

$$L \cdot I'' + R \cdot I' + C^{-1} \cdot I = 0$$

$$\dot{ES} = k_f \cdot E \cdot S - k_r \cdot ES$$

$$x'' + ax' + x + ax^2 = c_3$$

Programmable Analog Devices



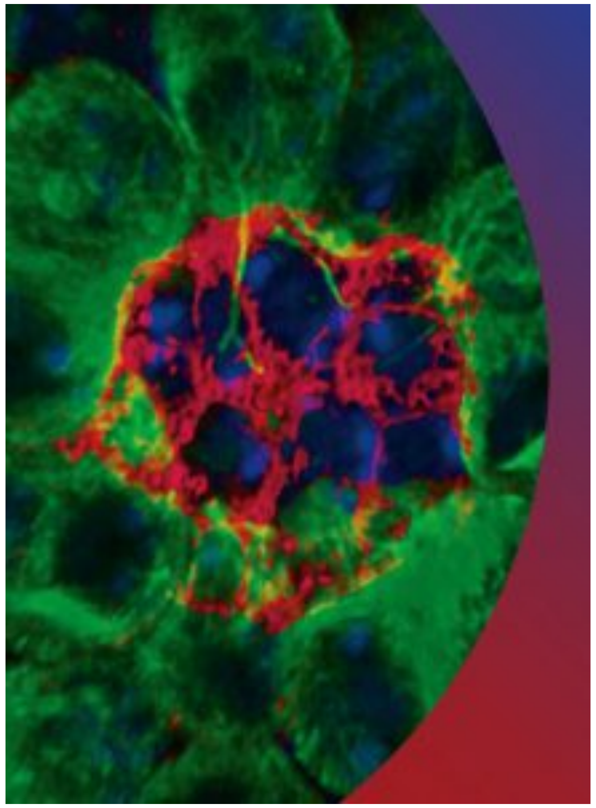
DYNAMICAL SYSTEMS MODEL THE PHYSICAL WORLD

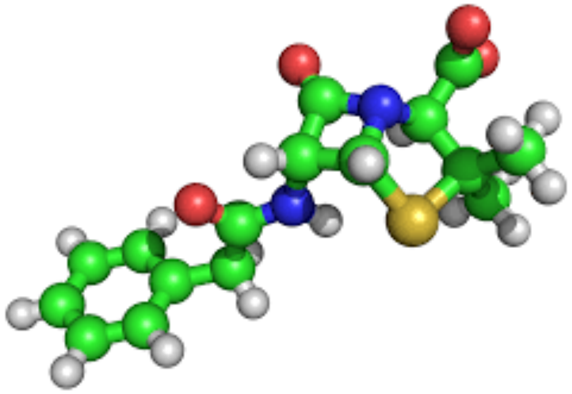
$$x'' + c_1 x' + c_2 x'' = 0$$

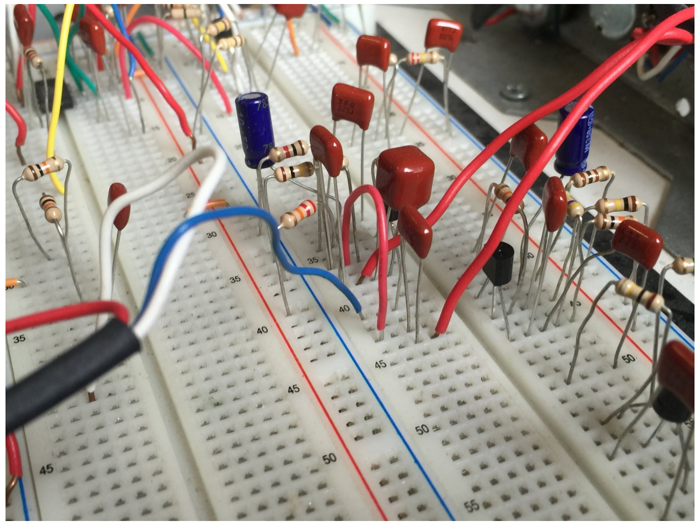
$$L \cdot I'' + R \cdot I' + C^{-1} \cdot I = 0$$

$$\dot{ES} = k_f \cdot E \cdot S - k_r \cdot ES$$

$$x'' + ax' + x + ax^2 = c_3$$







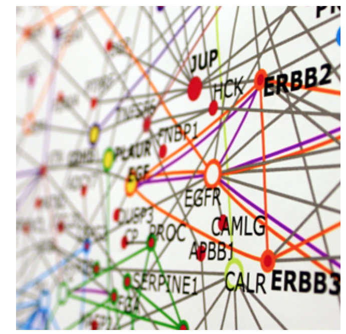
DYNAMICAL SYSTEMS MODEL BIOLOGICAL PROCESSES

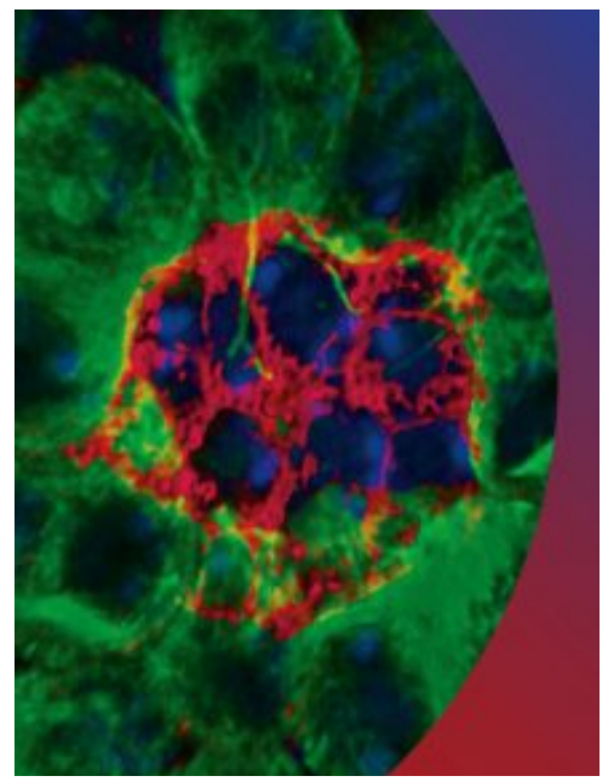
$$x'' + c_1 x' + c_2 x'' = 0$$

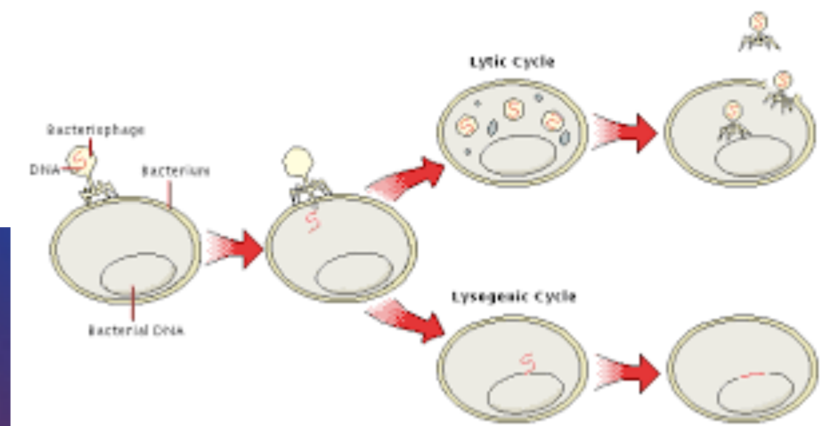
$$L \cdot I'' + R \cdot I' + C^{-1} \cdot I = 0$$

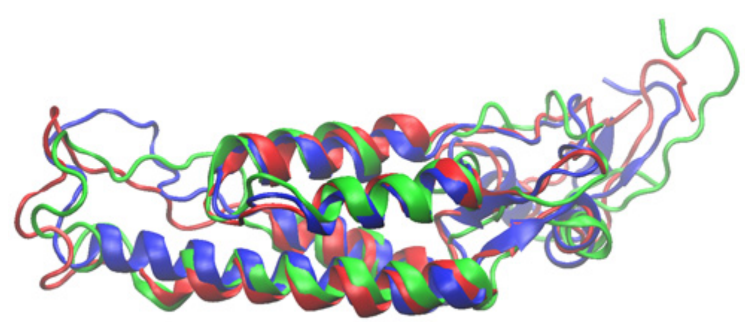
$$\dot{ES} = k_f \cdot E \cdot S - k_r \cdot ES$$

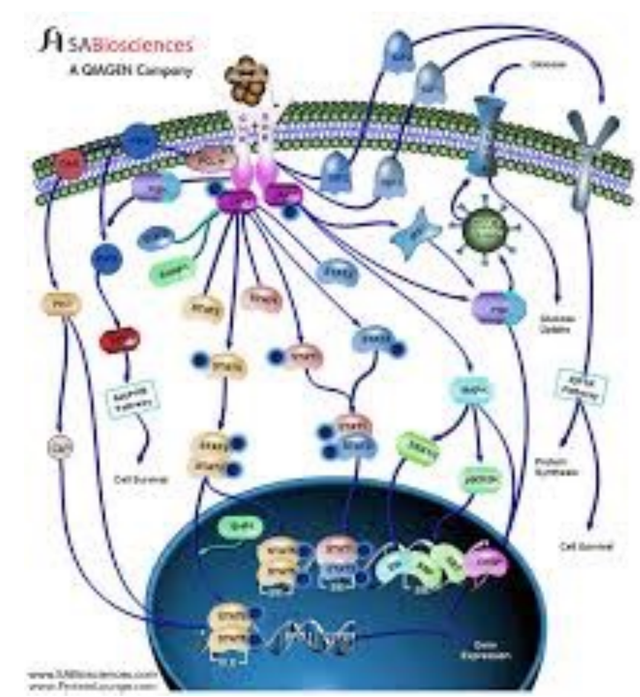
$$x'' + ax' + x + ax^2 = c_3$$



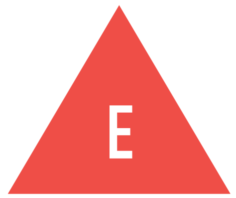








BIOLOGICAL DYNAMICAL SYSTEMS



$$E = E_0 - ES$$

$$S = S_0 - ES$$

$$\dot{ES} = k_f \cdot E \cdot S - k_r \cdot ES$$

state variables model physical quantities

BIOLOGICAL DYNAMICAL SYSTEMS

$$E = E_0 - ES$$

$$S = S_0 - ES$$

$$\dot{ES} = k_f \cdot E \cdot S - k_r \cdot ES$$

differential equations specify continuous dynamics of state variables over time

GOAL: SIMULATING BIOLOGICAL DYNAMICAL SYSTEMS

$$E = E_0 - ES \qquad E_0 = 6800$$

$$S = S_0 - ES \qquad S_0 = 4400$$

$$ES(0) = 0 \qquad \dot{E}S = k_f \cdot E \cdot S - k_r \cdot ES$$

given initial state of system: compute values of state variables over time

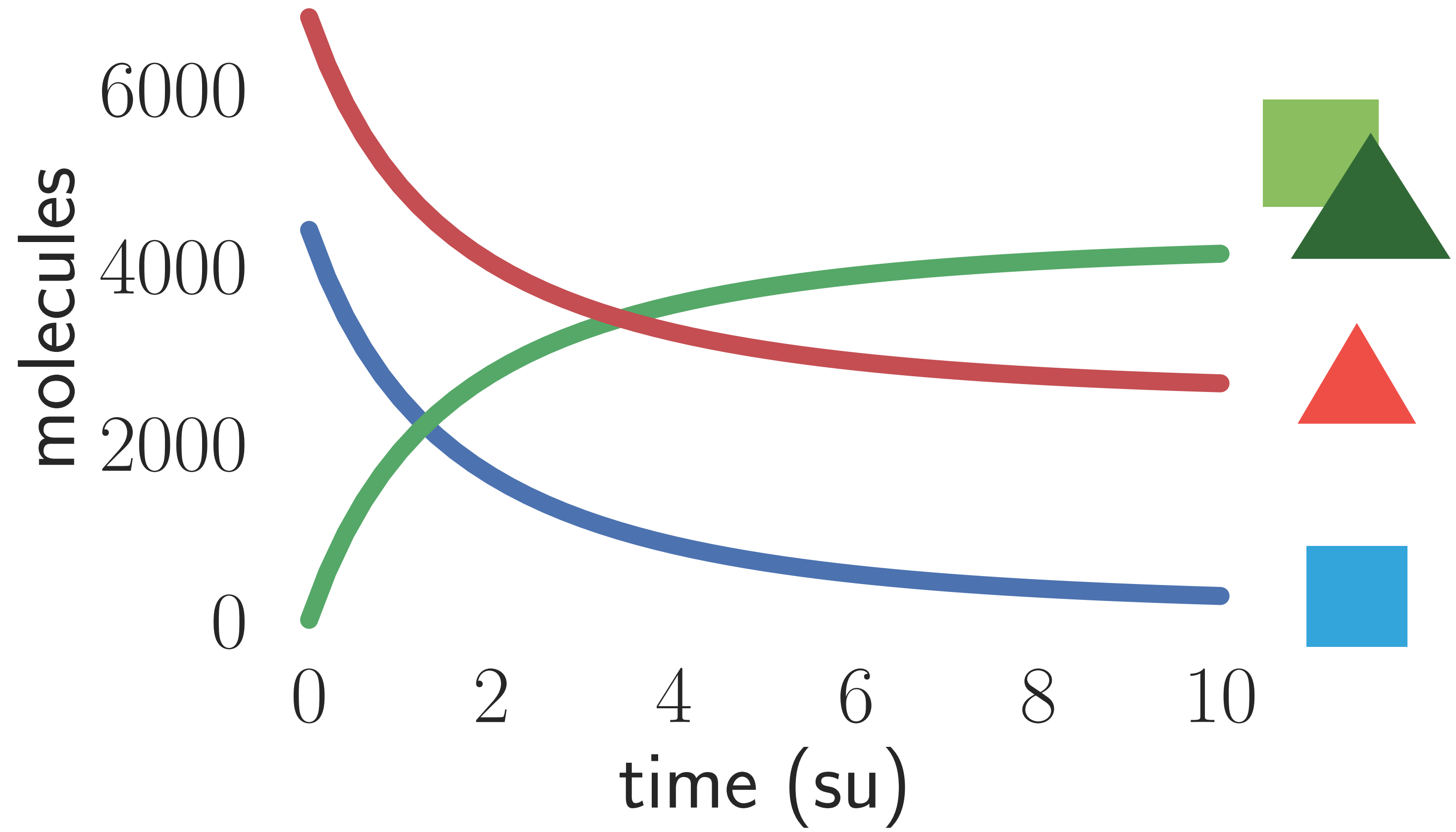
GOAL: SIMULATING BIOLOGICAL DYNAMICAL SYSTEMS

$E_0 = 6800 \quad E = E_0 - ES$

$S_0 = 4400 \quad S = S_0 - ES$

$\dot{ES} = k_f \cdot E \cdot S - k_r \cdot ES$

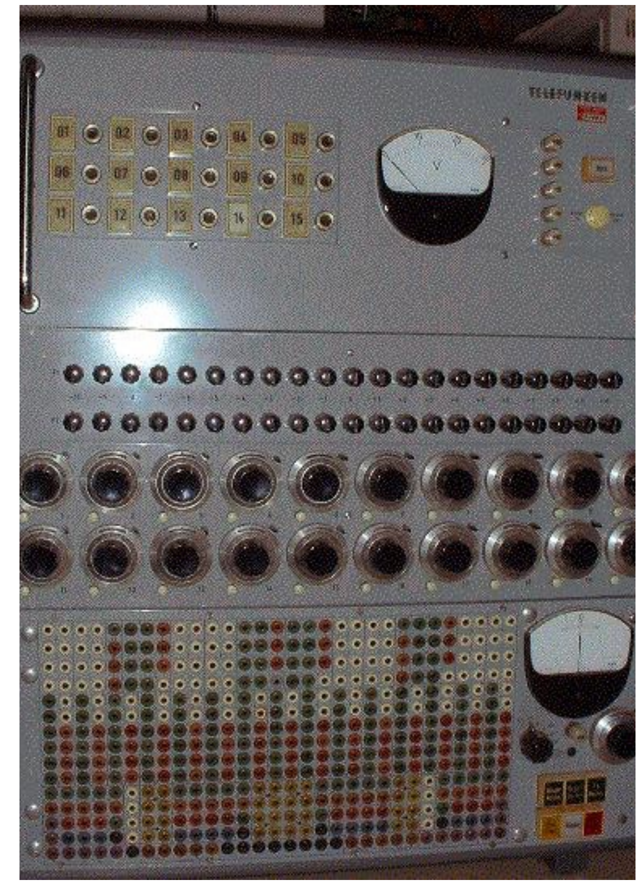
$ES(0) = 0$

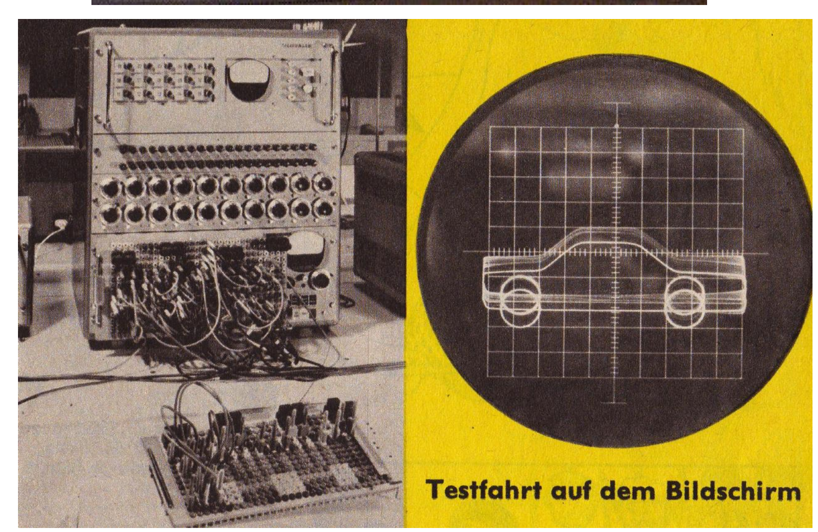


plot molecule counts/concentrations of compounds over time

ANALOG COMPUTING CIRCA 1950

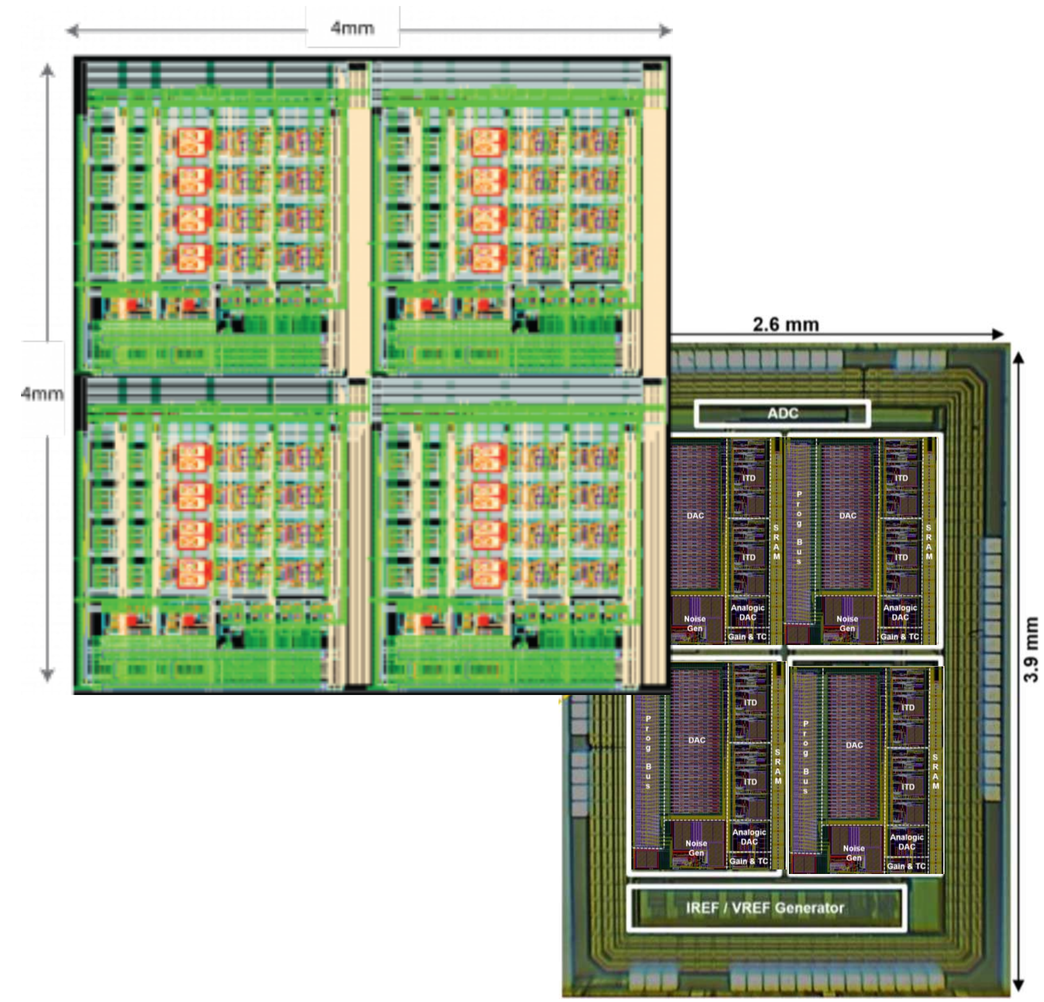
- **<u>direct mapping</u>**
  - variables → current, voltage
  - dynamics → circuit physics
- **<u>straightforward simulation</u>**
  - power up circuit
  - measure current, voltage over time
- **<u>1970-2010</u>**: Age of Digital Computing
  - Analog computes out of fashion





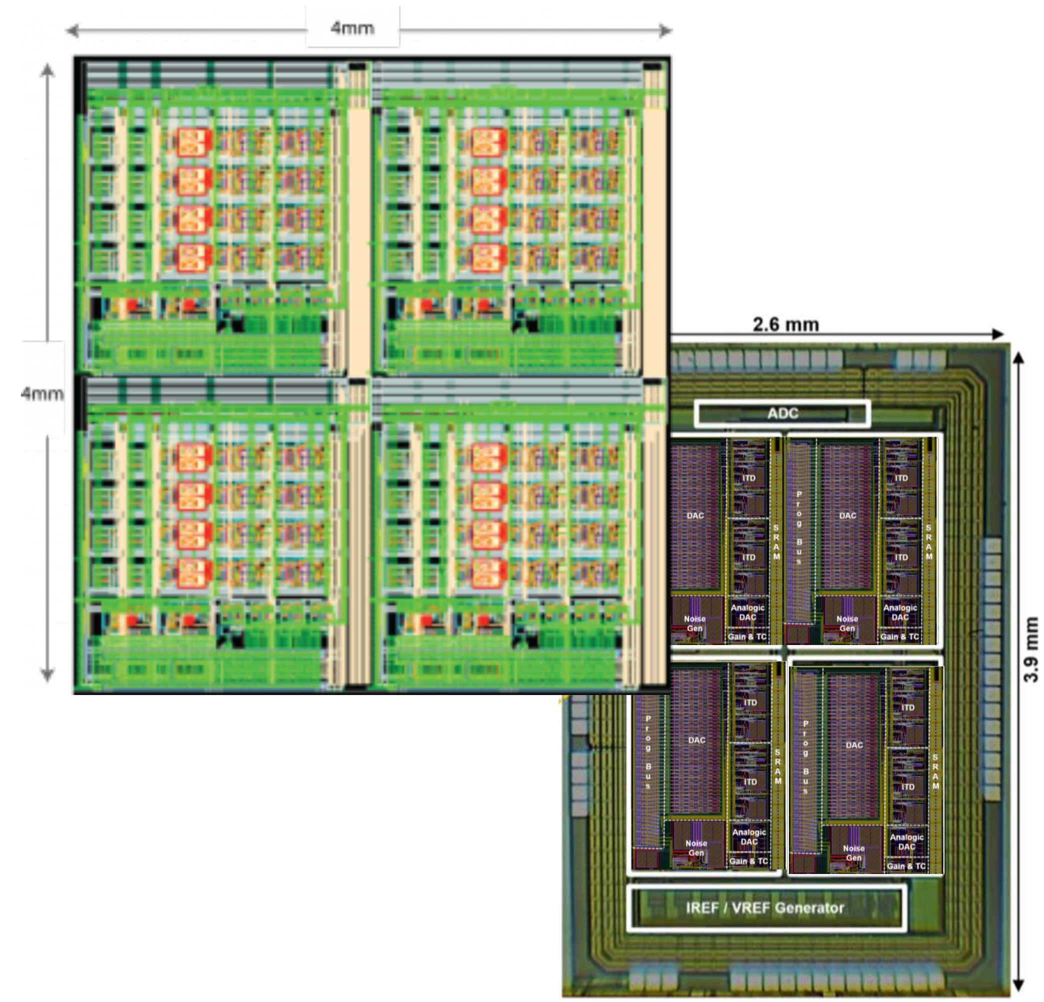
PROGRAMMABLE ANALOG DEVICES

- **<u>same computational model</u>**
- **<u>modernized hardware</u>**
  - modern semiconductor technologies
- **<u>new capabilities</u>**
  - powerful, heavily optimized building blocks
  - digital reprogrammability
  - exploit analog noise



PROGRAMMING CHALLENGES FOR ANALOG DEVICES

- **physical behavior**
  - voltage/current operating ranges
  - circuit noise [thermal/shot/flicker]
- **complex building blocks**
  - non-linear, non-convex
- **space limitations**
  - limitations on number of available blocks and connections
- **requires creativity when configuring device**



A COMPILER FOR PROGRAMMABLE ANALOG DEVICES

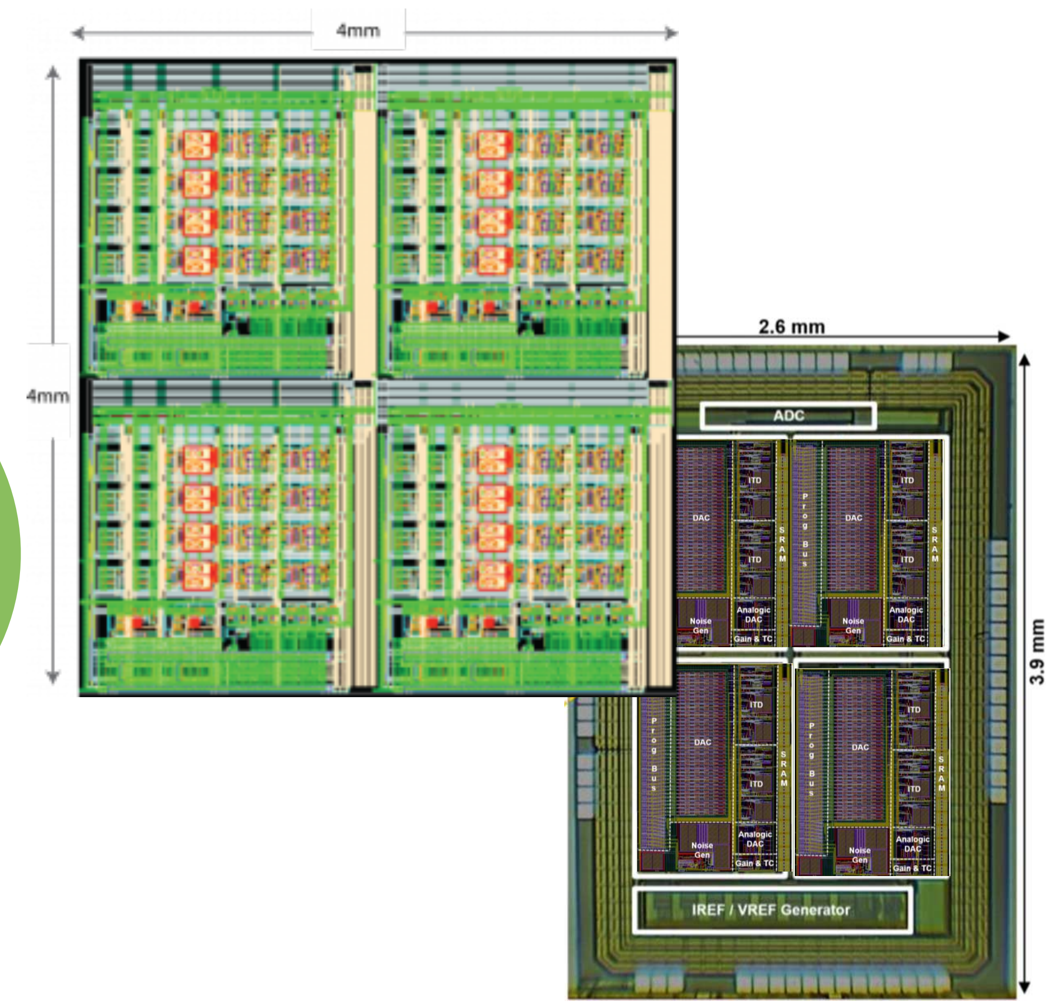
$$x'' + c_1 x' + c_2 x'' = 0$$

$$L \cdot I'' + R \cdot I' + C^{-1} \cdot I = 0$$

$$\dot{ES} = k_f \cdot E \cdot S - k_r \cdot ES$$

$$x'' + ax' + x + ax^2 = c_3$$

Dynamical Systems



Programmable
Analog Devices

automatically compose complex algebraic building blocks

automatically reason about operating ranges + circuit noise

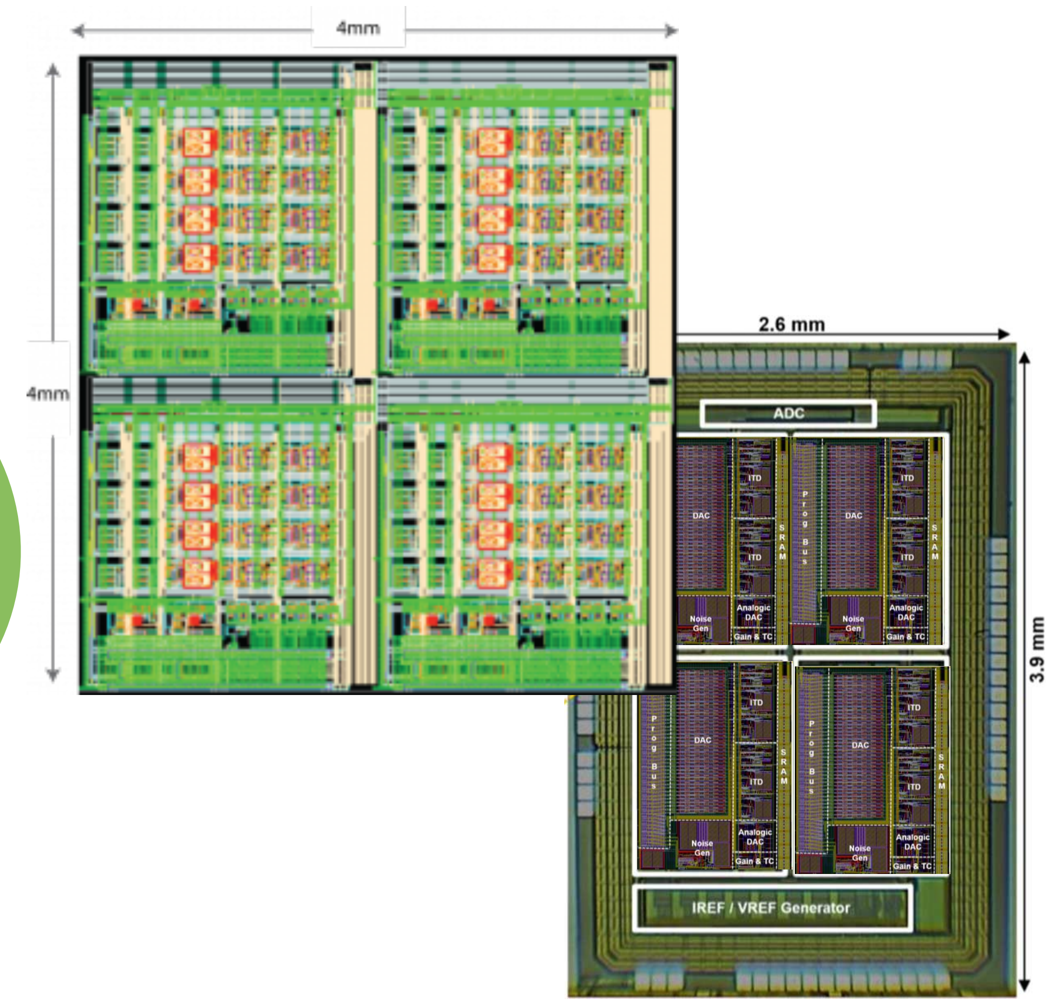
$$x'' + c_1 x' + c_2 x'' = 0$$

$$L \cdot I'' + R \cdot I' + C^{-1} \cdot I = 0$$

$$\dot{E}S = k_f \cdot E \cdot S - k_r \cdot ES$$

$$x'' + ax' + x + ax^2 = c_3$$

Dynamical Systems



Programmable
Analog Devices

automatically compose complex algebraic building blocks

automatically reason about operating ranges + circuit noise

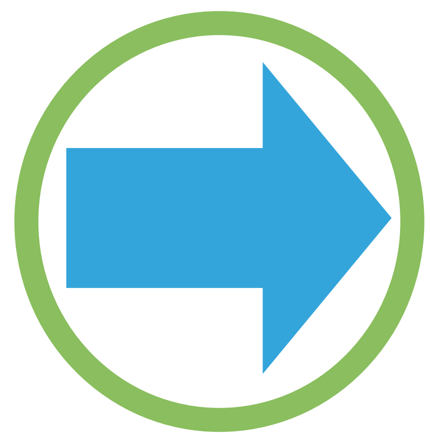
$$x'' + c_1 x' + c_2 x'' = 0$$

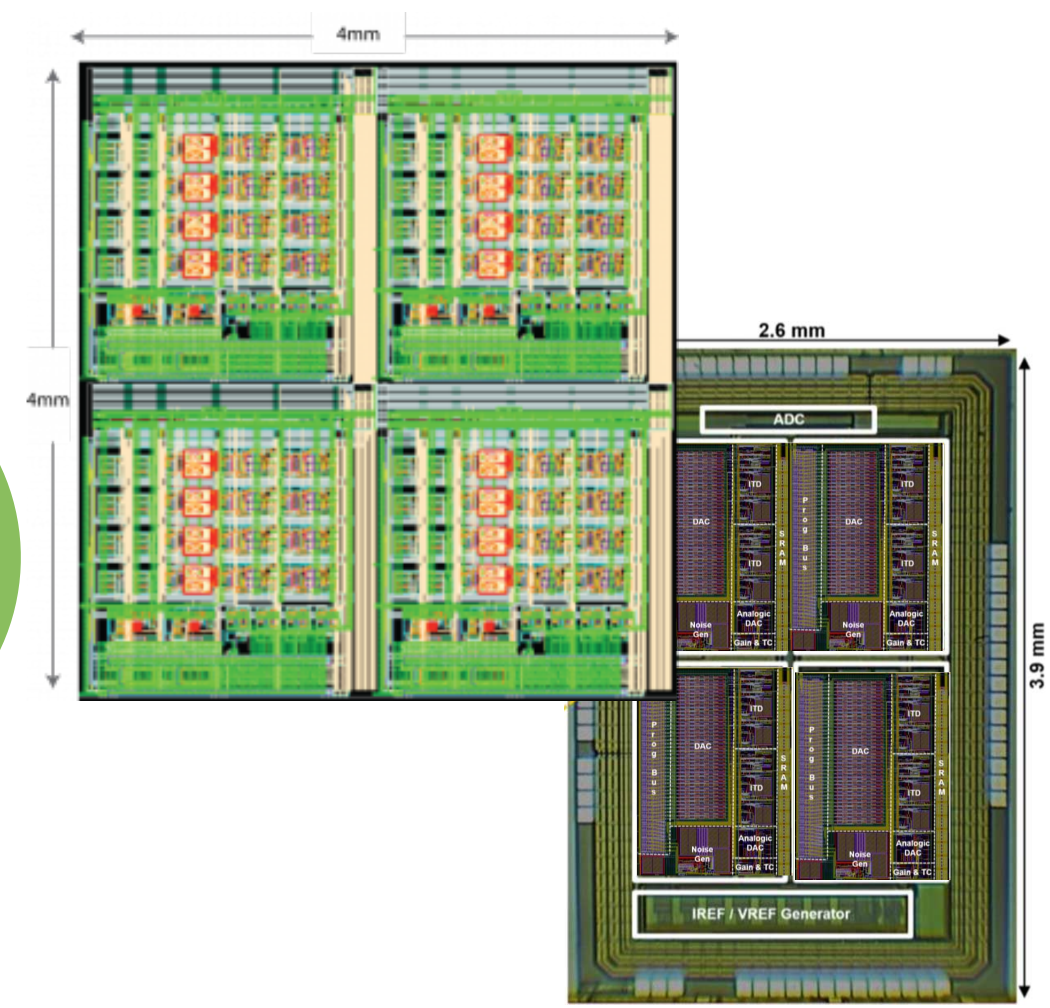
$$L \cdot I'' + R \cdot I' + C^{-1} \cdot I = 0$$

$$\dot{ES} = k_f \cdot E \cdot S - k_r \cdot ES$$

$$x'' + ax' + x + ax^2 = c_3$$

Dynamical Systems





Programmable
Analog Devices

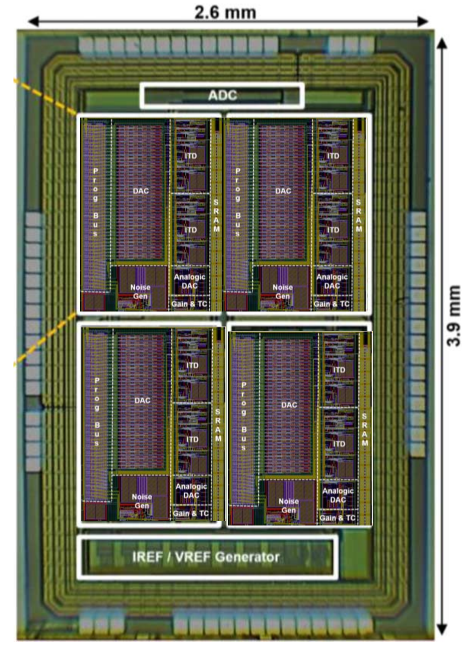
FUNDAMENTALLY NEW COMPILATION TECHNIQUES

TALK OUTLINE

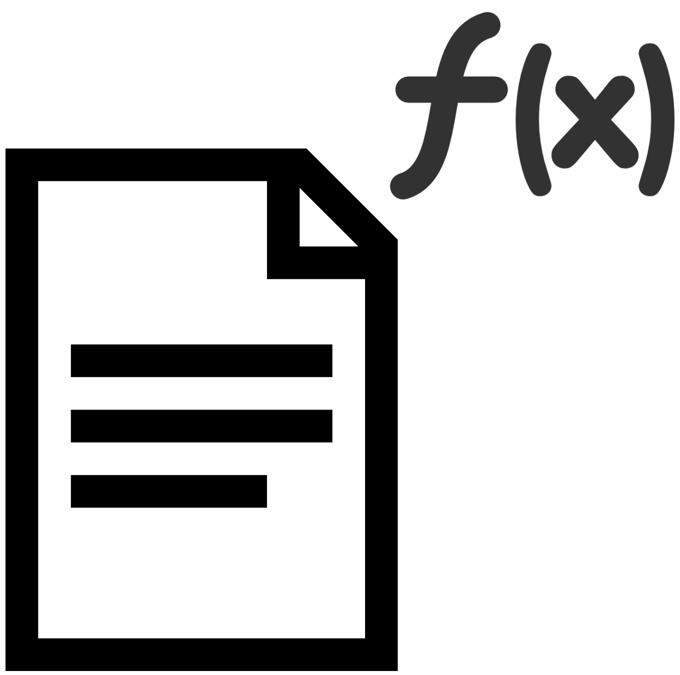
- Background: overview of compilation problem
- Arco Compiler[1]: automatically configure analog devices to simulate dynamical systems.
- Jaunt Solver[2]: automatically scales dynamical systems to execute analog hardware with operating range constraints.
- Closing Remarks

1. Configuration Synthesis for Programmable Analog Devices with Arco. Sara Achour, Rahul Sarpeshkar and Martin Rinard. June 2016. PLDI 2016.
2. Time Dilation and Contraction for Programmable Analog Devices with Jaunt. Sara Achour and Martin Rinard. December 2017. ASPLOS 2018.

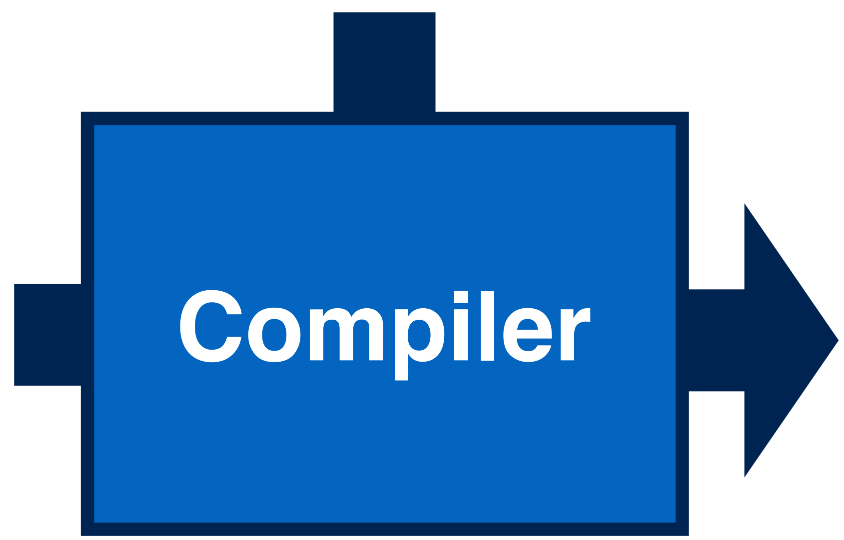
BACKGROUND

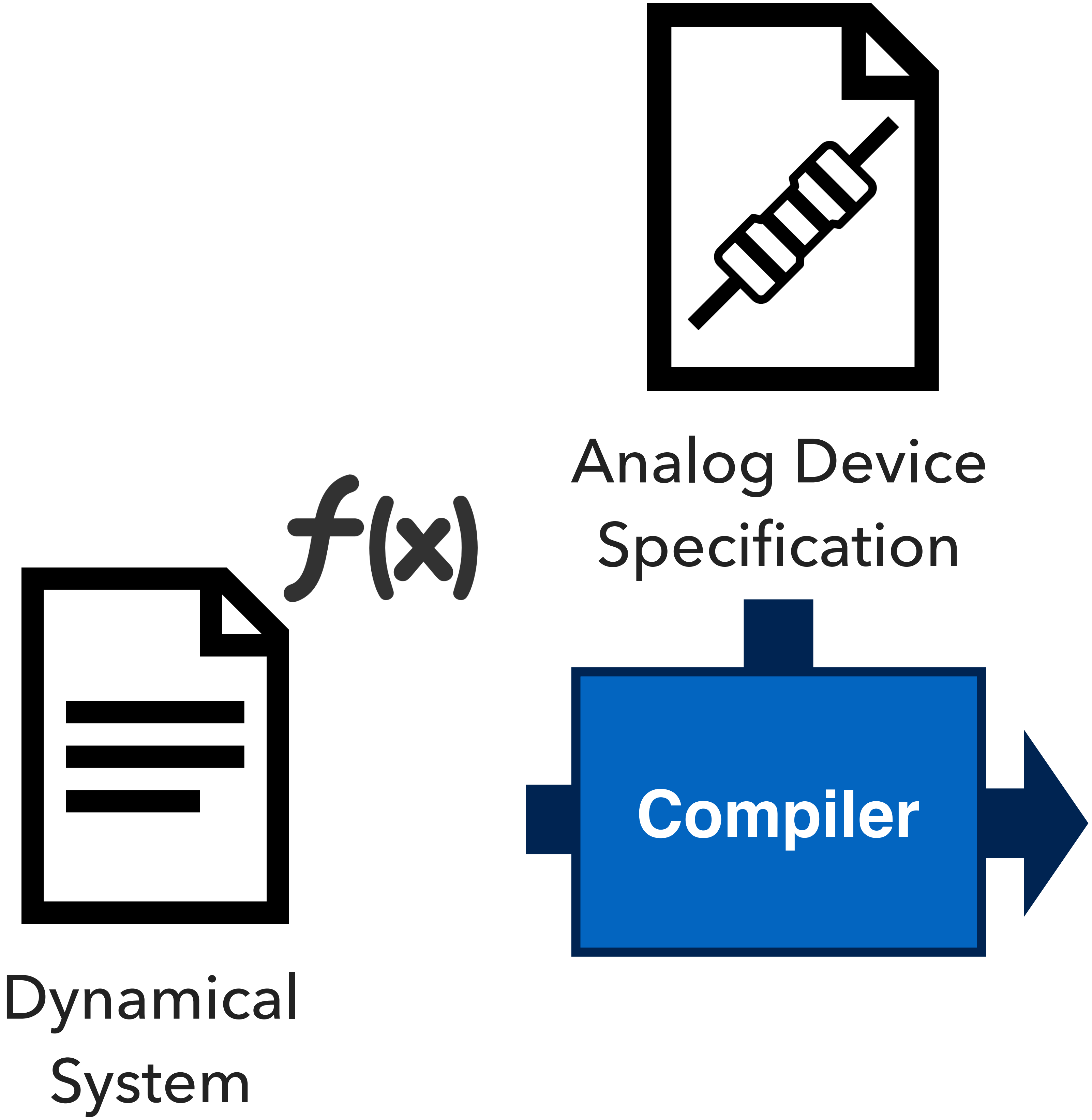


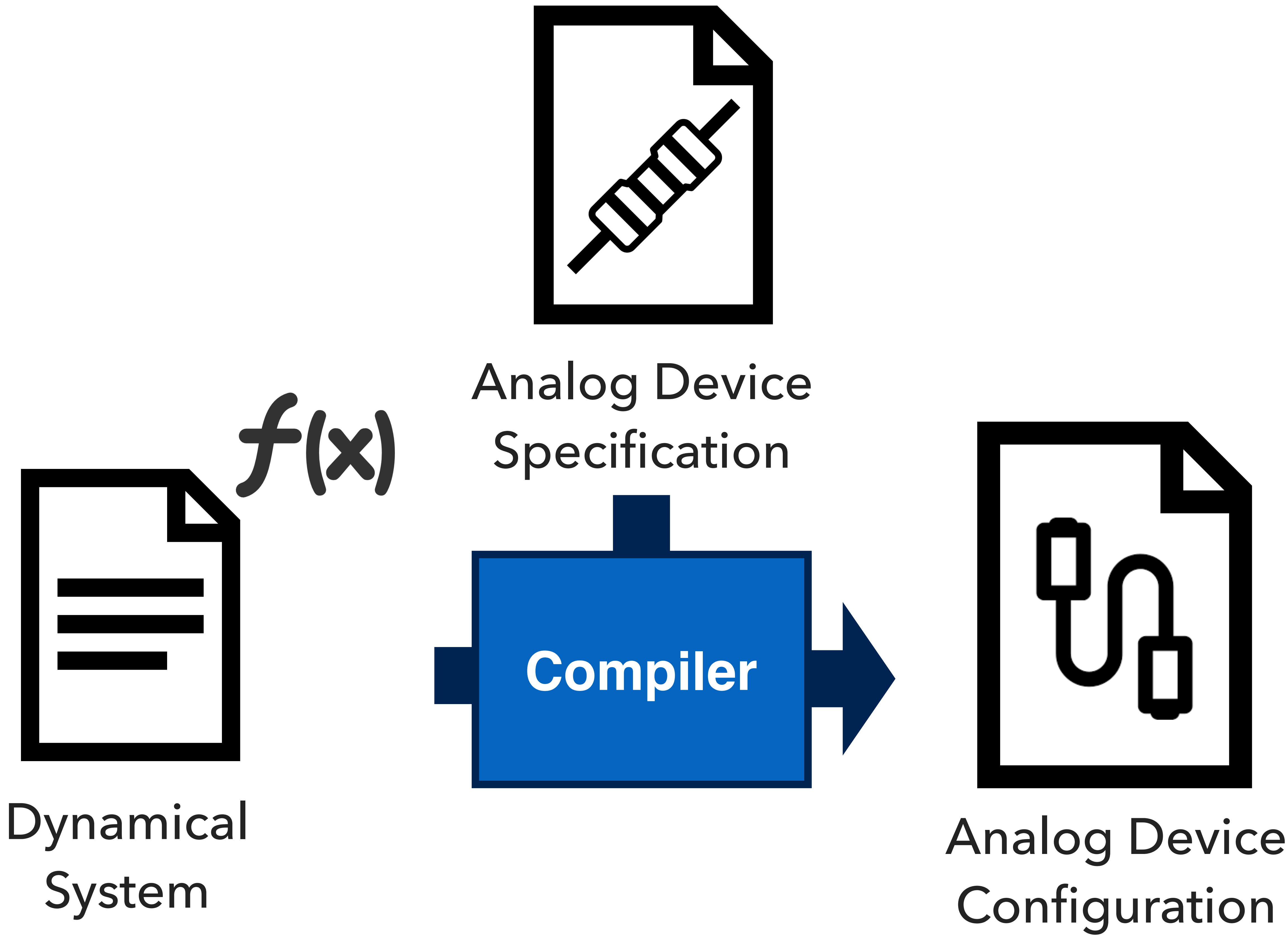
Analog Device

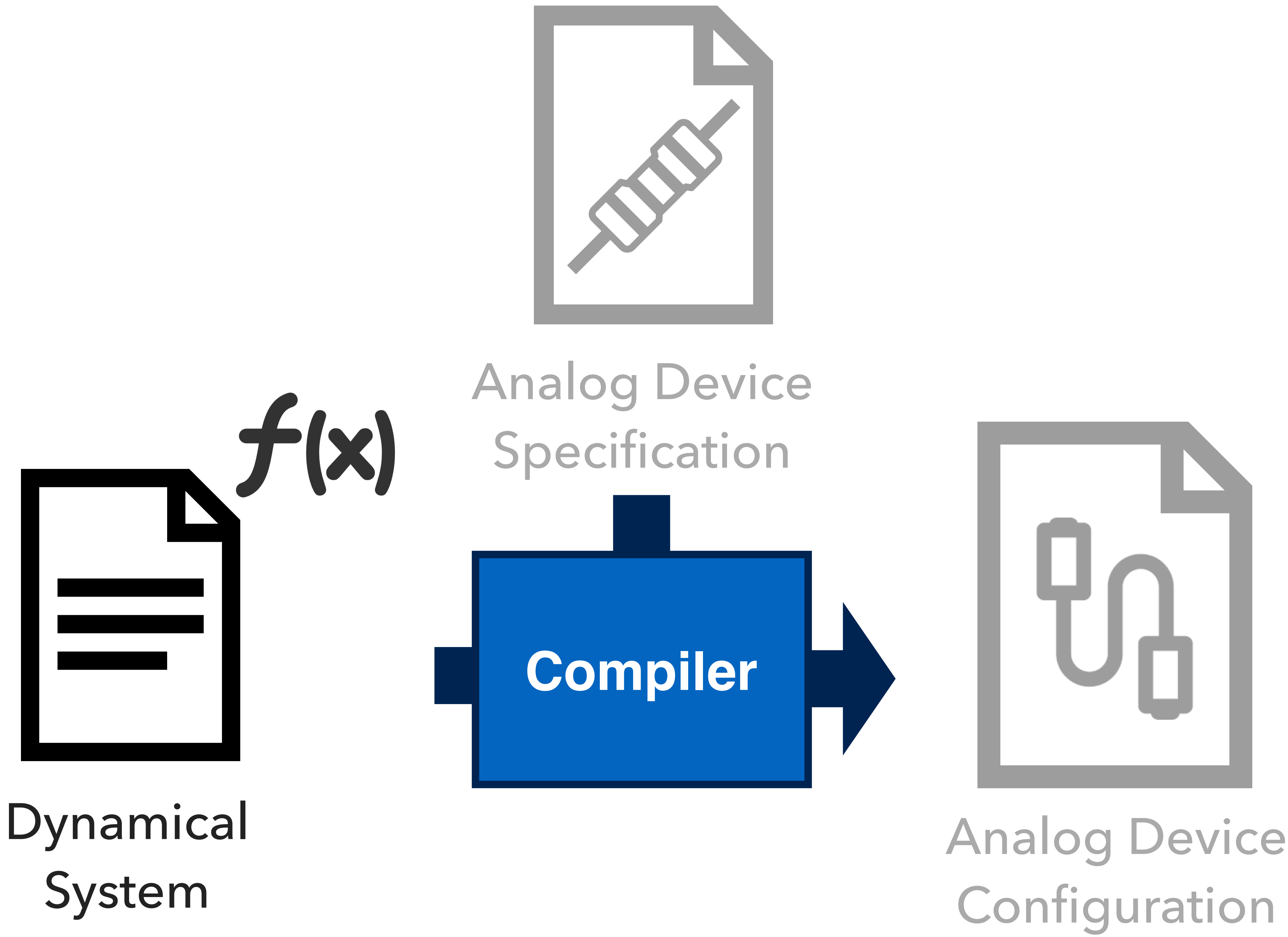


Dynamical System




Analog Device
Specification
Compiler
Dynamical
System


Analog Device
Specification
Dynamical
System
Compiler
Analog Device
Configuration


Analog Device
Specification
Dynamical
System
Compiler
Analog Device
Configuration

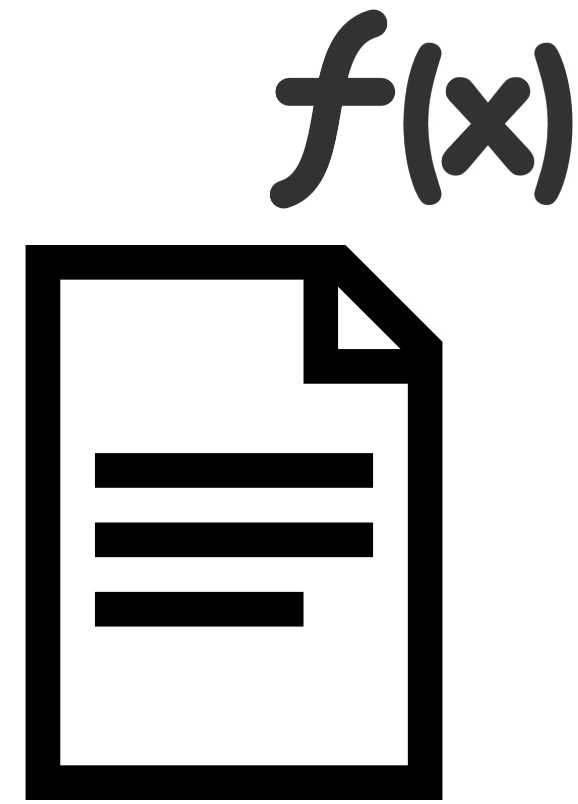
DYNAMICAL SYSTEM SPECIFICATION

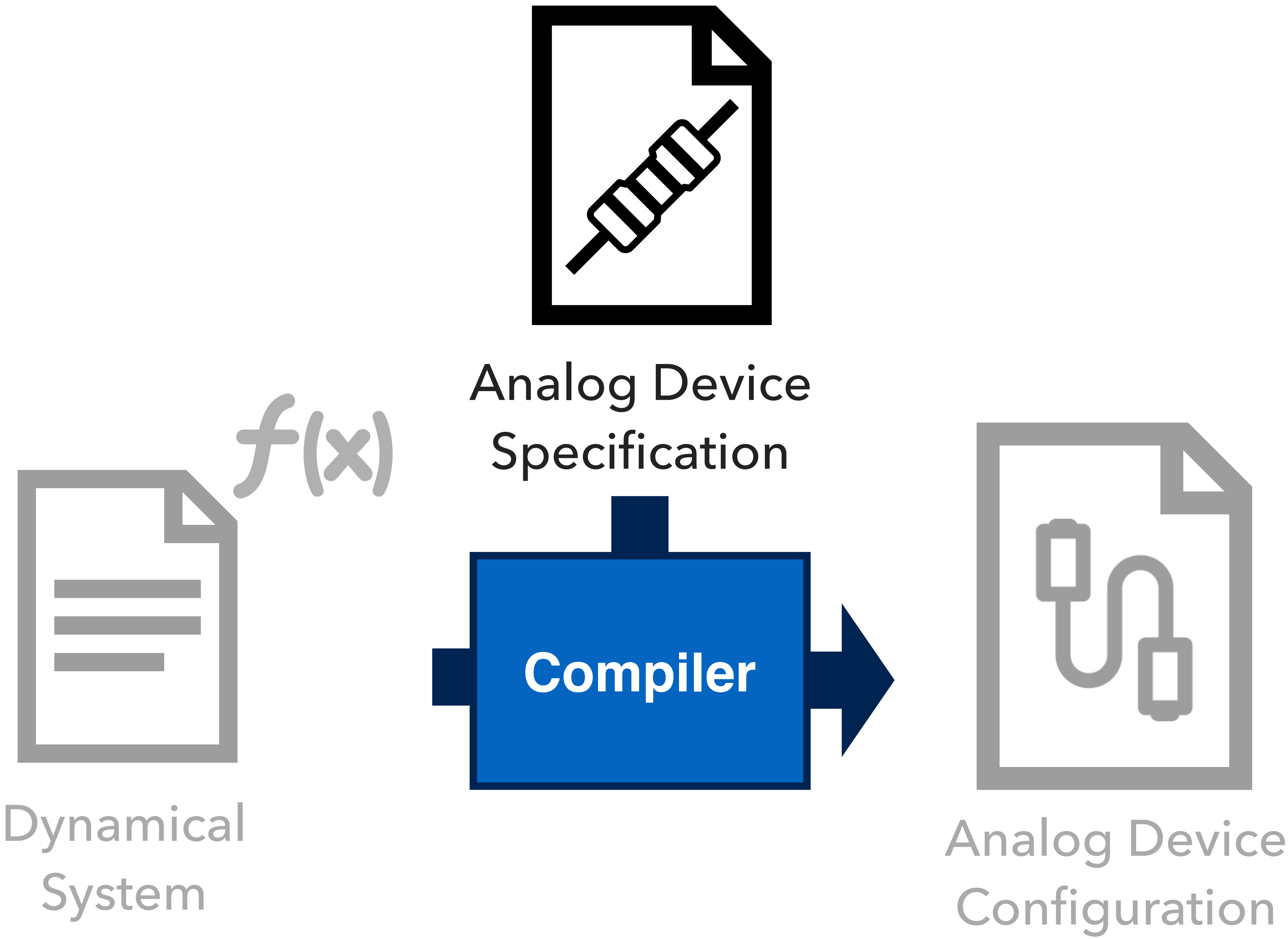
$$E = 6800 - ES$$

$$S = 4400 - ES$$

$$\dot{ES} = 10^{-4} \cdot E \cdot S - 10^{-2} \cdot ES$$

$$ES(0) = 0$$




Analog Device
Specification
f(x)
Compiler
Dynamical
System
Analog Device
Configuration

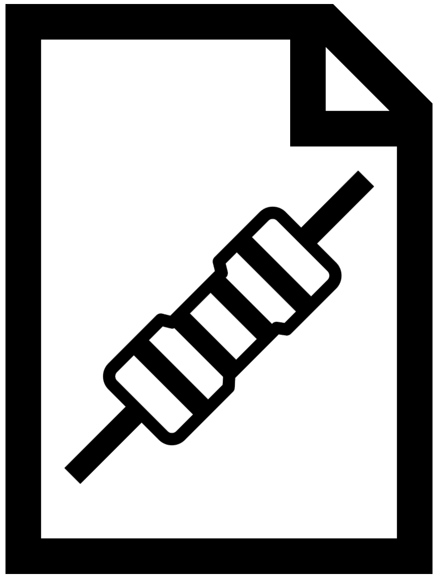
ANALOG DEVICE SPECIFICATION

MM x 3

DAC x 5

IADD x 3

ADC x 5



ANALOG DEVICE SPECIFICATION

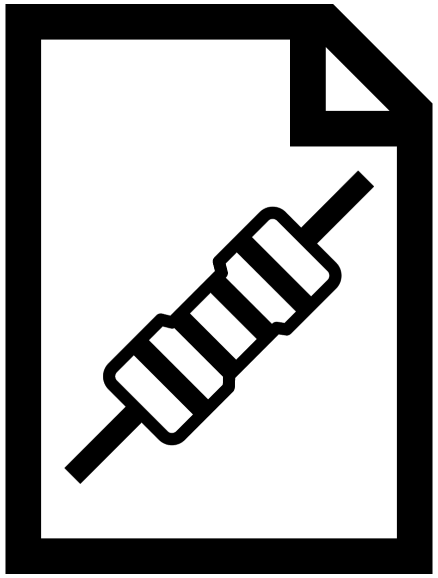
x 3

inp X0, Y0, Z0

inp A, B

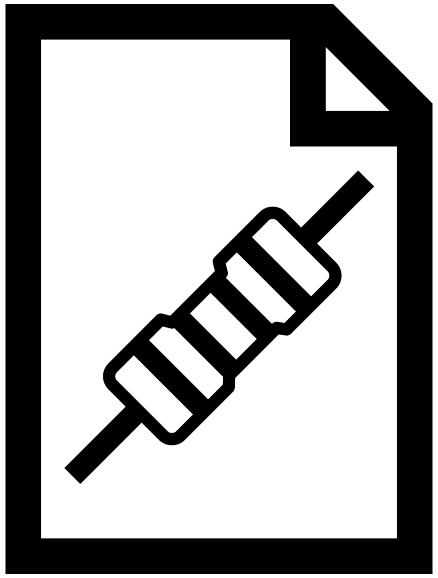
out X, Y, Z

Input and Output Ports



ANALOG DEVICE SPECIFICATION

x 3



inp X0, Y0, Z0

inp A, B

out X, Y, Z

rel X.I = X0.I - Z.I

rel Y.I = Y0.I - Z.I

rel Z.I′ = A.V X.I Y.I - B.V Z.I

and Z.I(0) = Z0

Block Dynamics

ANALOG DEVICE SPECIFICATION

DAC x 5

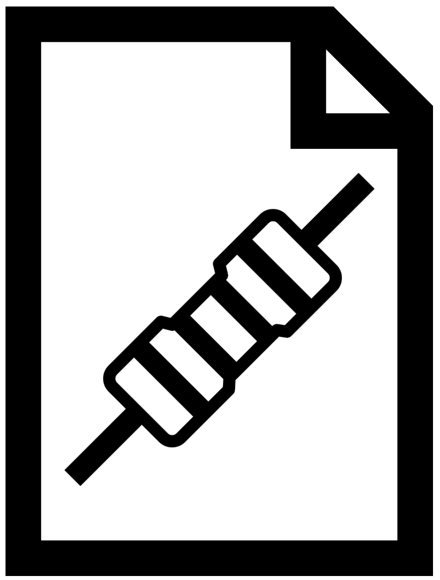
inp X **digital**
out Z

rel Z.I = X

ADC x 5

inp X
out Z **digital**

rel Z = X.I



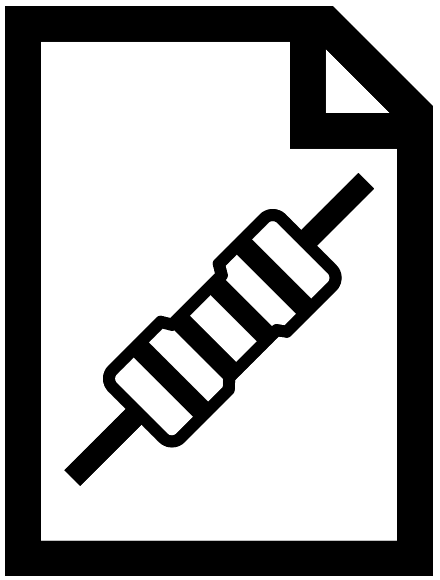
ANALOG DEVICE SPECIFICATION

MM x 3

DAC x 5

IADD x 3

ADC x 5

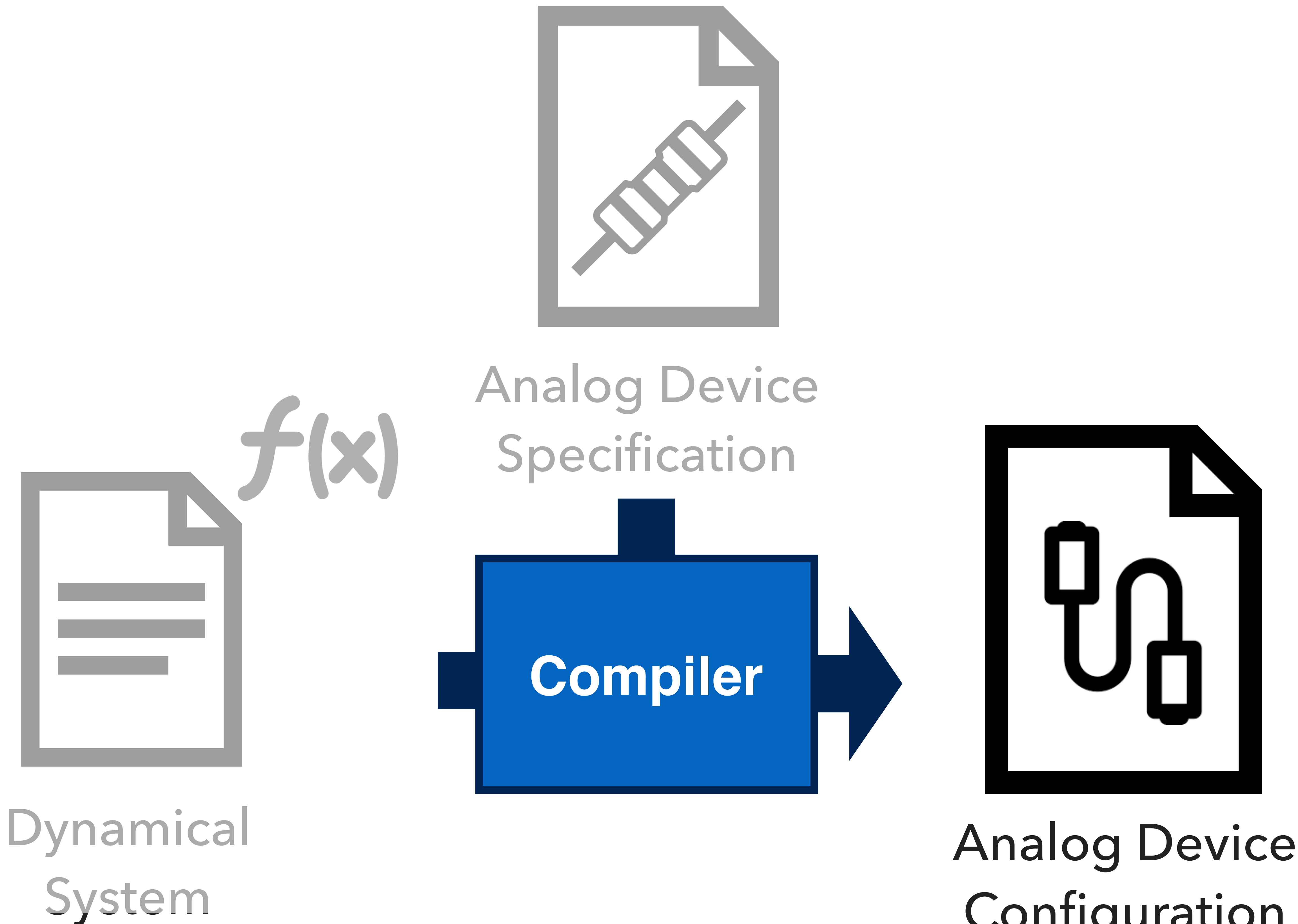


conn MM[*].Z **and** ADC[*].X

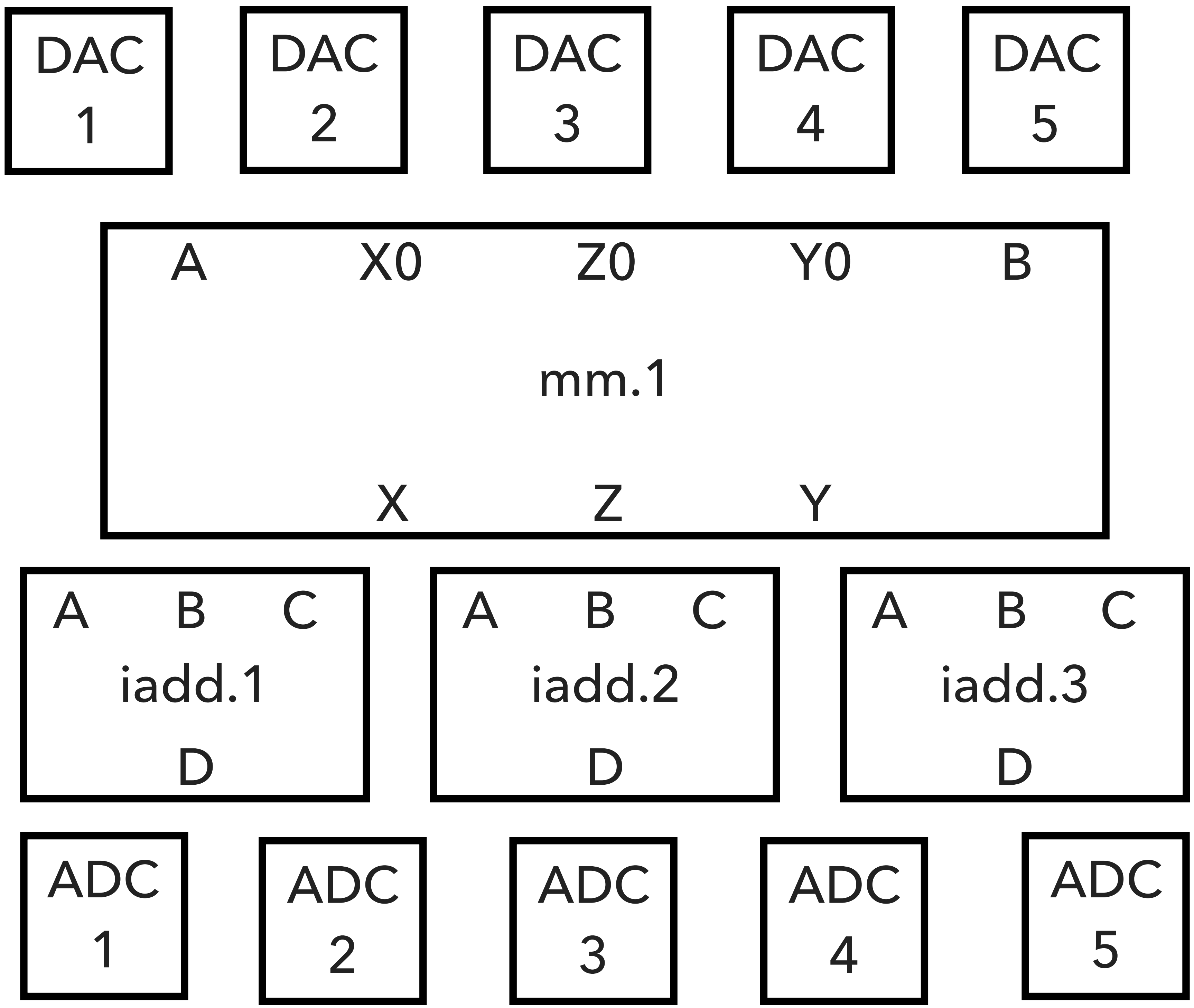
conn DAC[*].Z **and** MM[*].A

conn MM[1].Z **and** MM[2].X

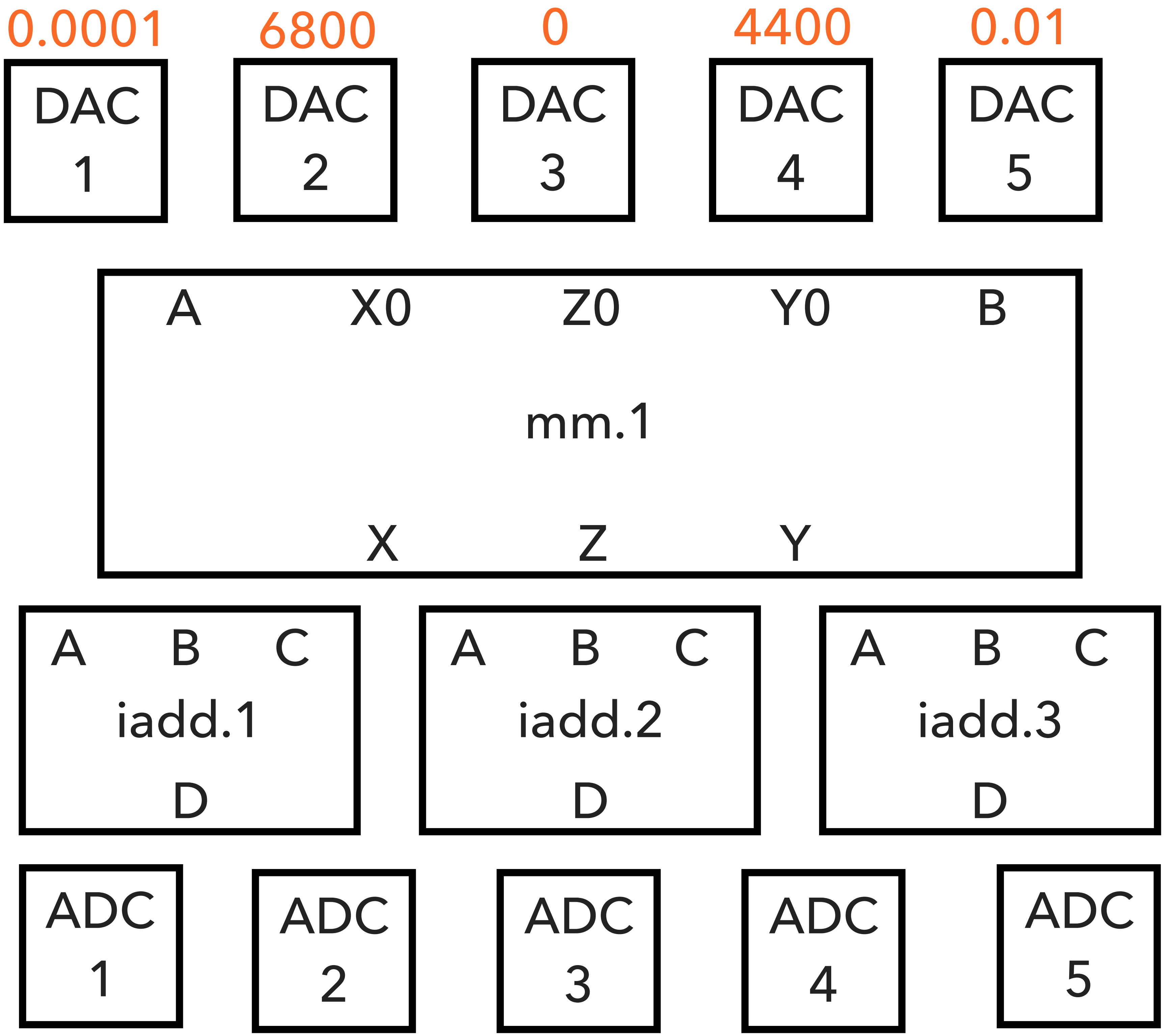
Available Connections

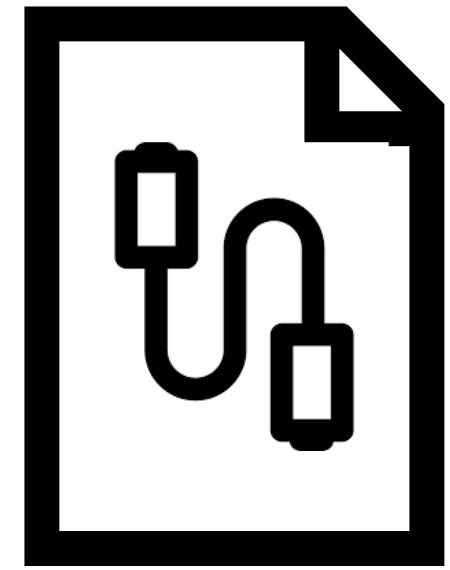

Analog Device
Specification
Dynamical
System
Compiler
Analog Device
Configuration

ANALOG DEVICE CONFIGURATION

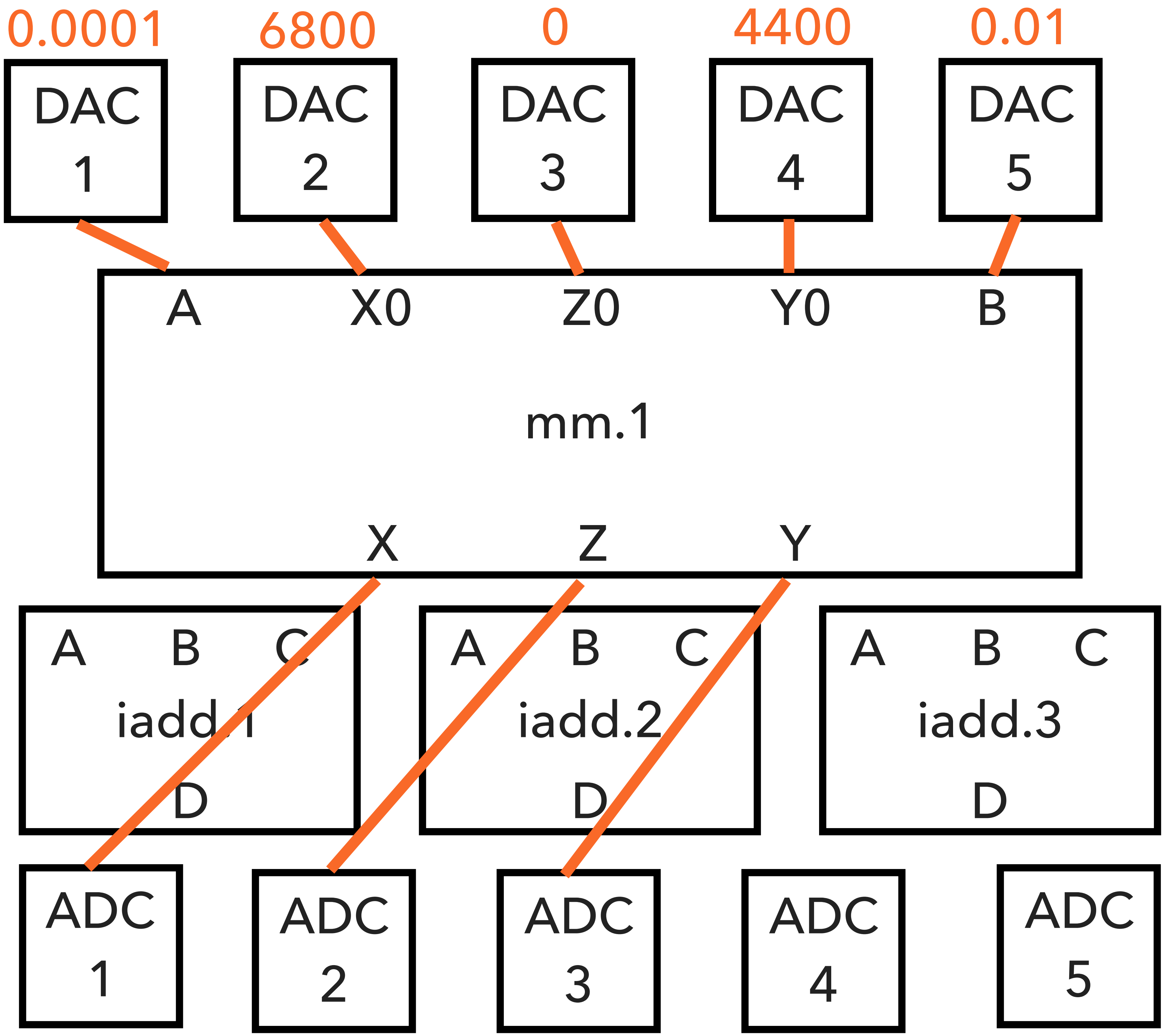


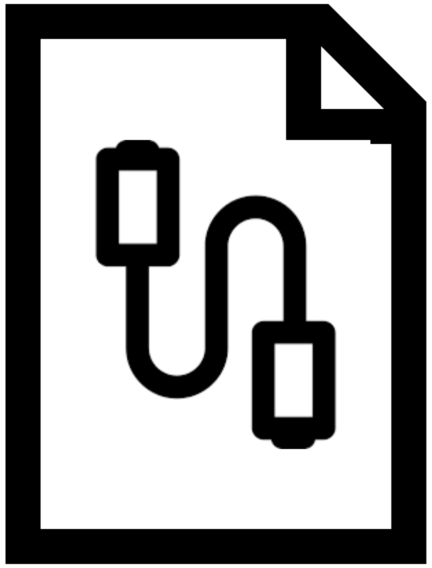
ANALOG DEVICE CONFIGURATION



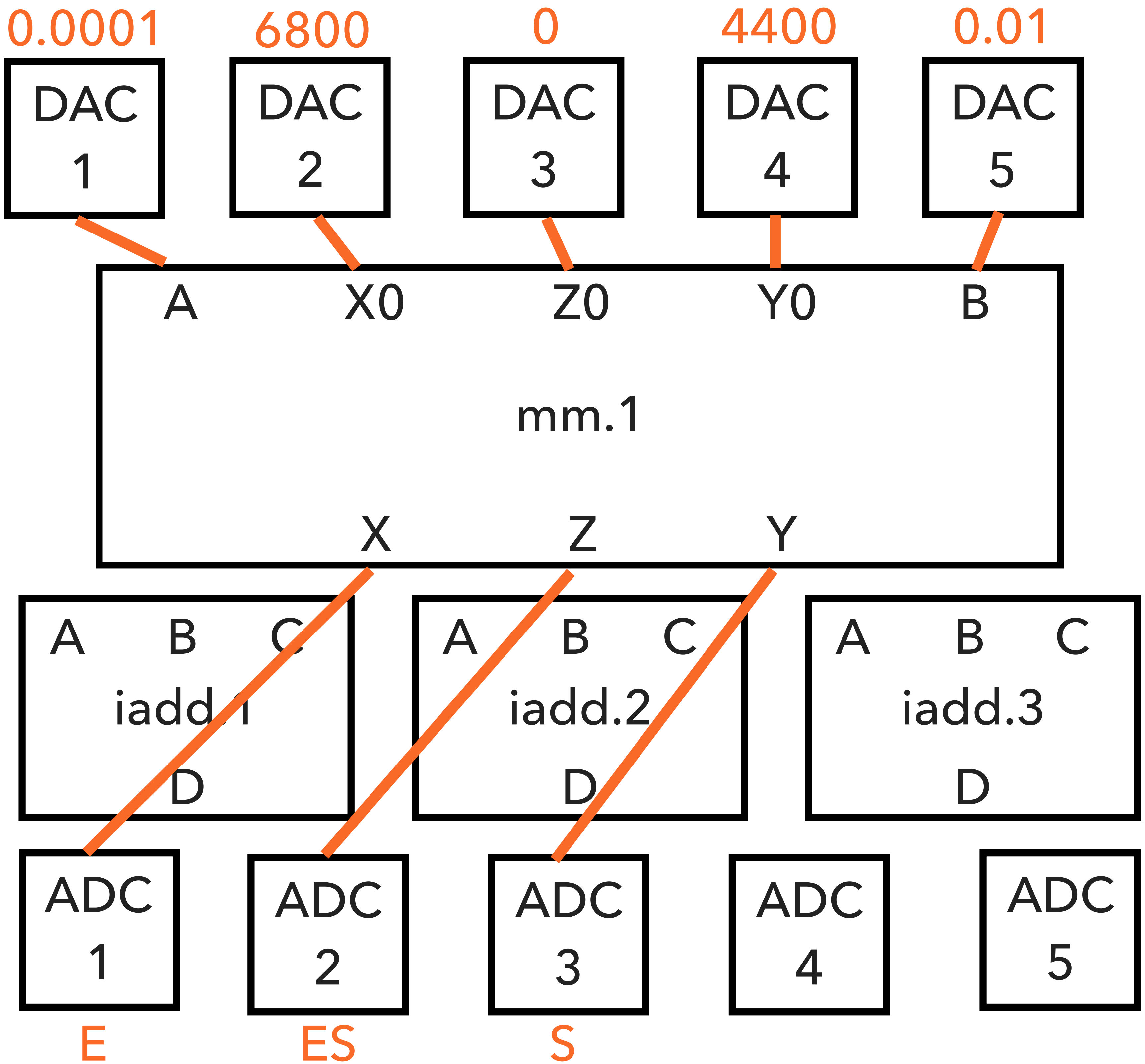


ANALOG DEVICE CONFIGURATION

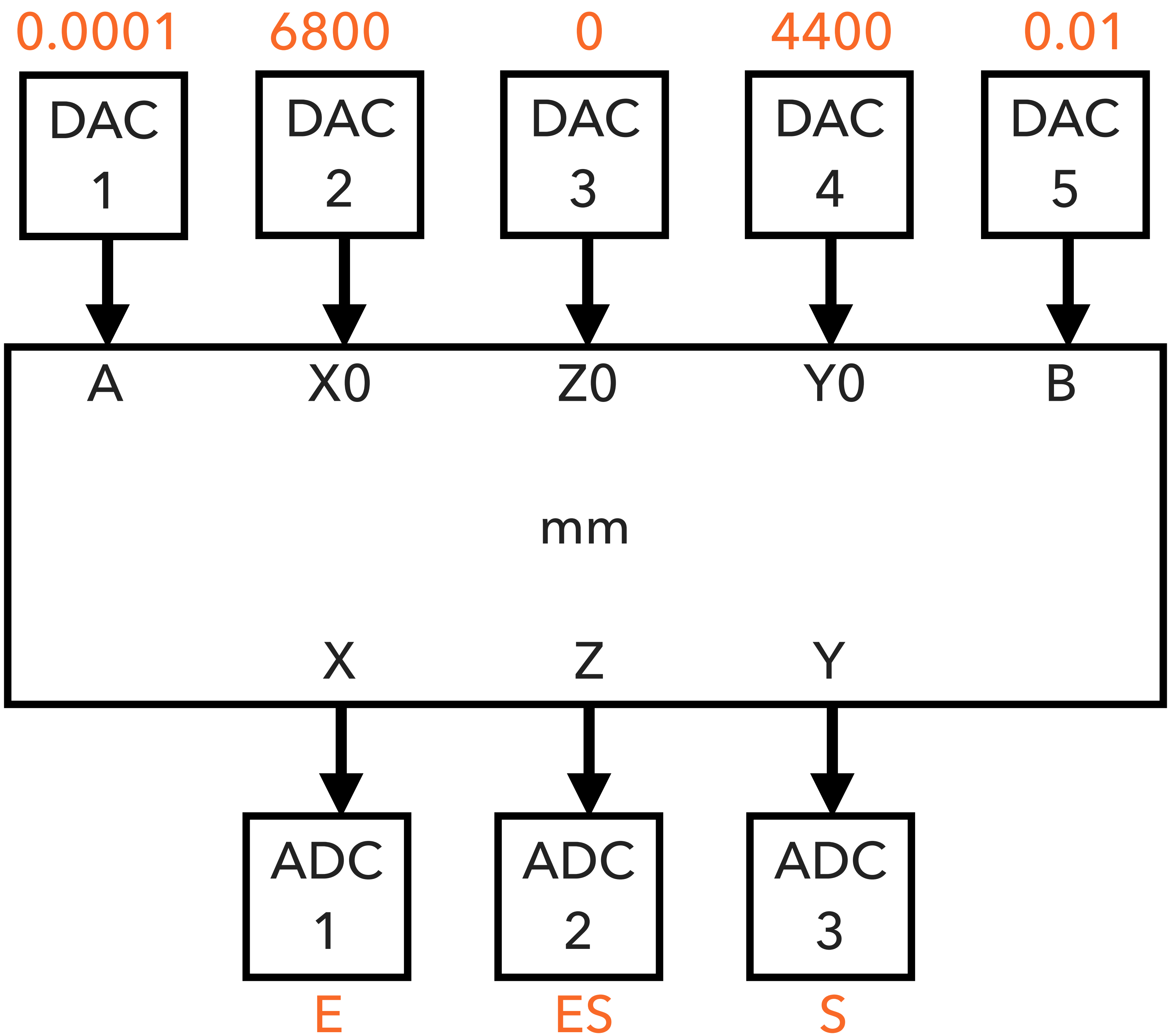


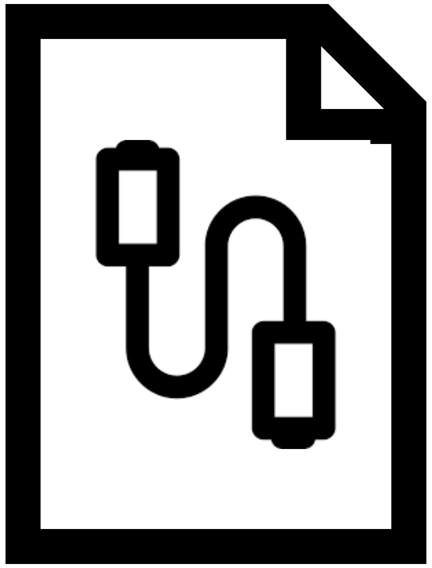


ANALOG DEVICE CONFIGURATION



ANALOG DEVICE CONFIGURATION

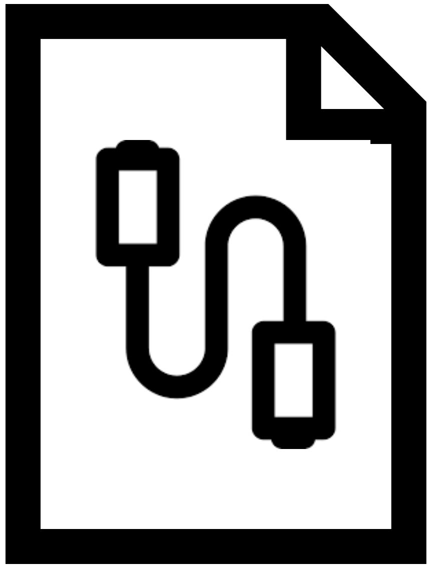




ANALOG DEVICE CONFIGURATION

```
set DAC[0].X = 0.003
    ...
set DAC[5].X = 0.006
```

DAC Values

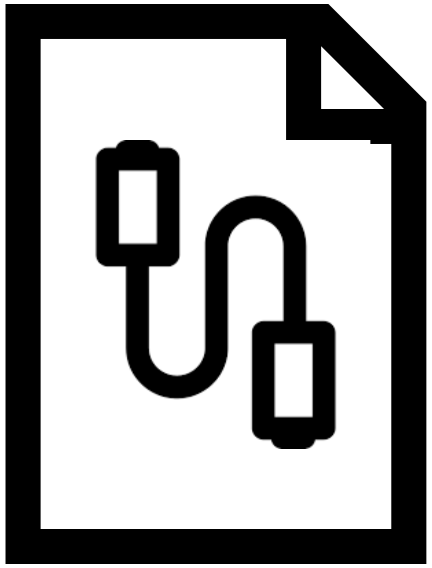


ANALOG DEVICE CONFIGURATION

set DAC[0].X = 0.003

…

set DAC[5].X = 0.006



lbl ADC[0].Z = "E"

lbl ADC[1].Z = "S"

lbl ADC[2].Z = "ES"

ADC Values

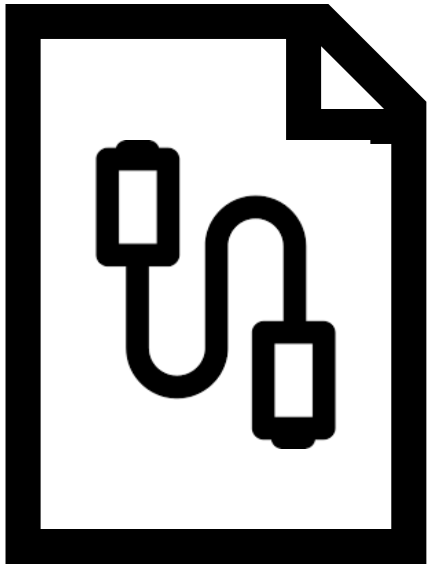
ANALOG DEVICE CONFIGURATION

set DAC[0].X = 0.003
…
set DAC[5].X = 0.006

lbl ADC[0].Z = “E”

lbl ADC[1].Z = “S”

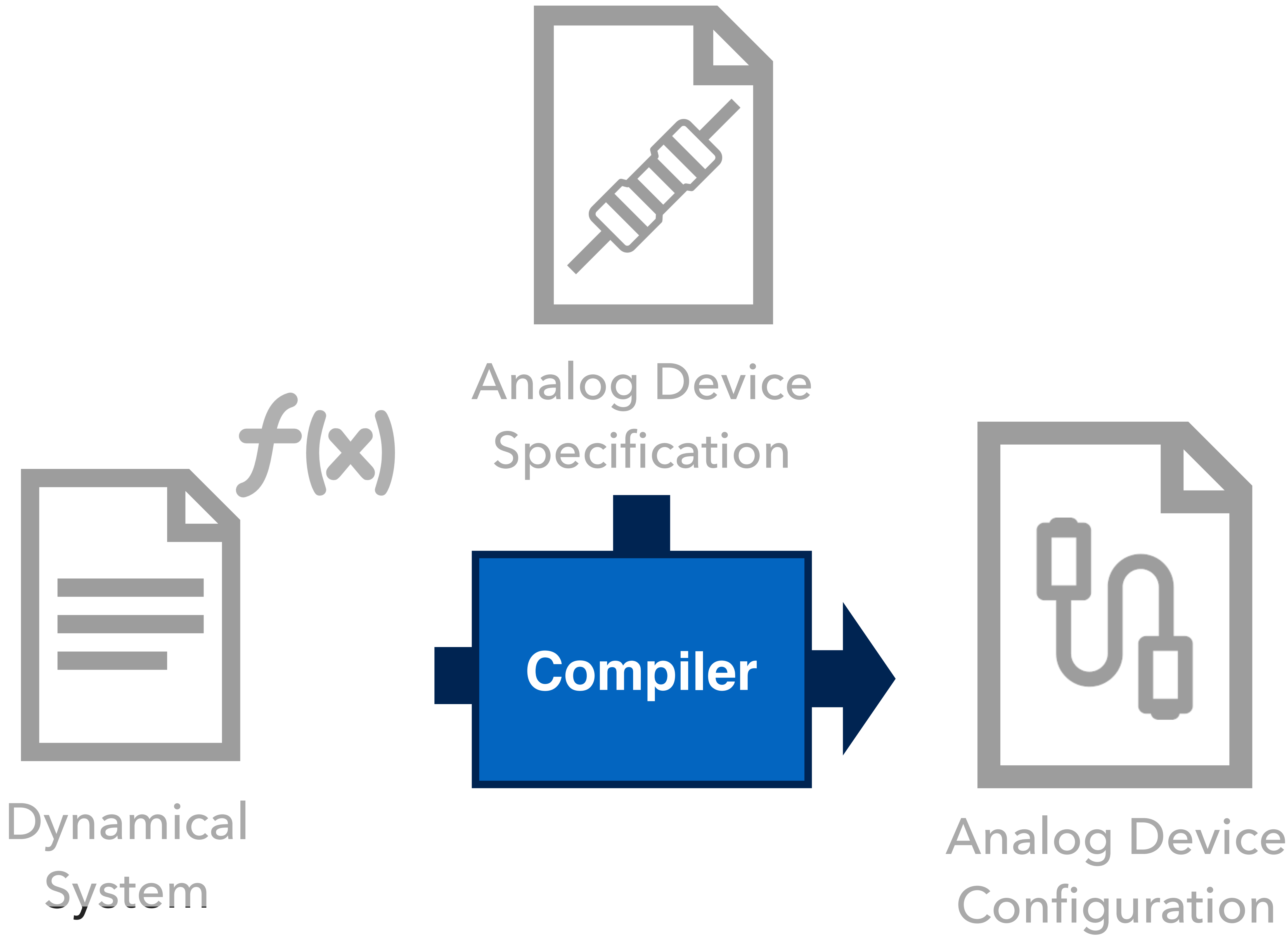
lbl ADC[2].Z = “ES”



conn DAC[0].Z **to** MM[0].X0
…
conn MM[0].Y **to** ADC[2].X

conn MM[0].Z **to** ADC[3].X

Connections


Analog Device
Specification
Dynamical
System
Compiler
Analog Device
Configuration

TALK OUTLINE

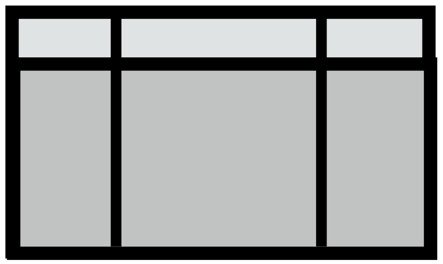
- ~~Background: overview of compilation problem~~
- Arco Compiler[1]: automatically configure analog devices to simulate dynamical systems.
- Jaunt Solver[2]: automatically scales dynamical systems to execute analog hardware with operating range constraints.
- Closing Remarks

1. Configuration Synthesis for Programmable Analog Devices with Arco. Sara Achour, Rahul Sarpeshkar and Martin Rinard. June 2016. PLDI 2016.
2. Time Dilation and Contraction for Programmable Analog Devices with Jaunt. Sara Achour and Martin Rinard. December 2017. ASPLOS 2018.

ARCO COMPILER

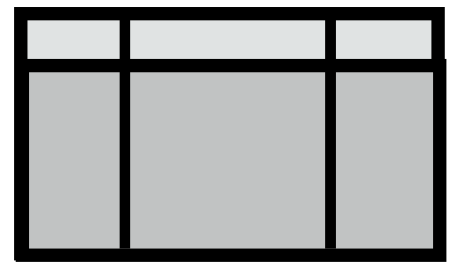
ARCO COMPILER OVERVIEW

Arco performs a search over **tableaus**



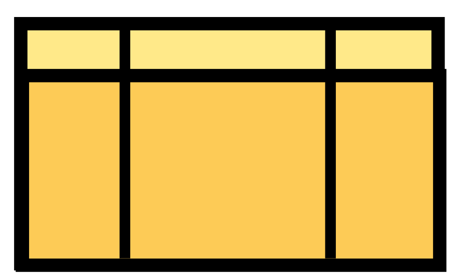
ARCO COMPILER

tableau : search state



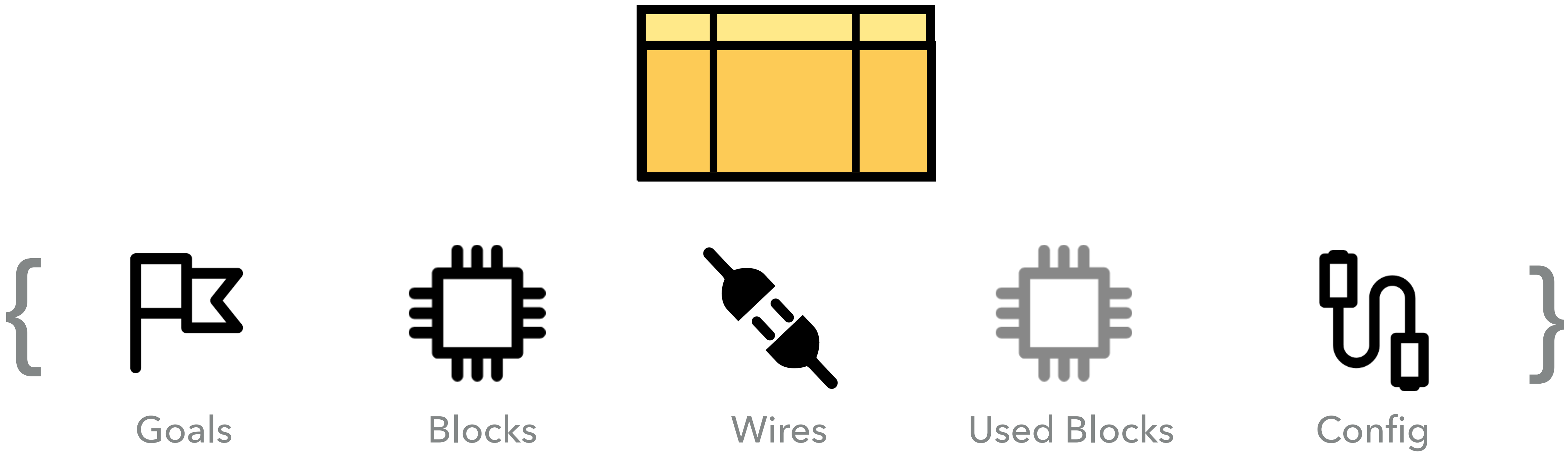
ARCO COMPILER OVERVIEW

Arco starts with an **initial tableau**



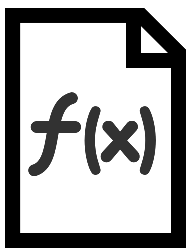
ARCO COMPILER

initial tableau : the initial state of the search



ARCO COMPILER

initial tableau : the initial state of the search



$$E = 6800 - ES$$

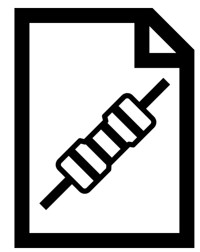
$$S = 4400 - ES$$

$$\dot{E}S = 10^{-4}E \cdot S - 10^{-2}ES$$

$$ES(0) = 0$$

ARCO COMPILER

initial tableau : the initial state of the search



$$E = 6800 - ES$$

$$S = 4400 - ES$$

$$\dot{E}S = 10^{-4}E \cdot S - 10^{-2}ES$$

$$ES(0) = 0$$

DAC[0]

ADC[0]

MM[0]

DAC[1]

$DAC[0].O$ — $MM[0].A$

$DAC[0].O$ — $MM[0].X0$

$MM[0].Z$ — $ADC[0].X$

$MM[0].X$ — $ADC[0].X$

ARCO COMPILER

initial tableau : the initial state of the search

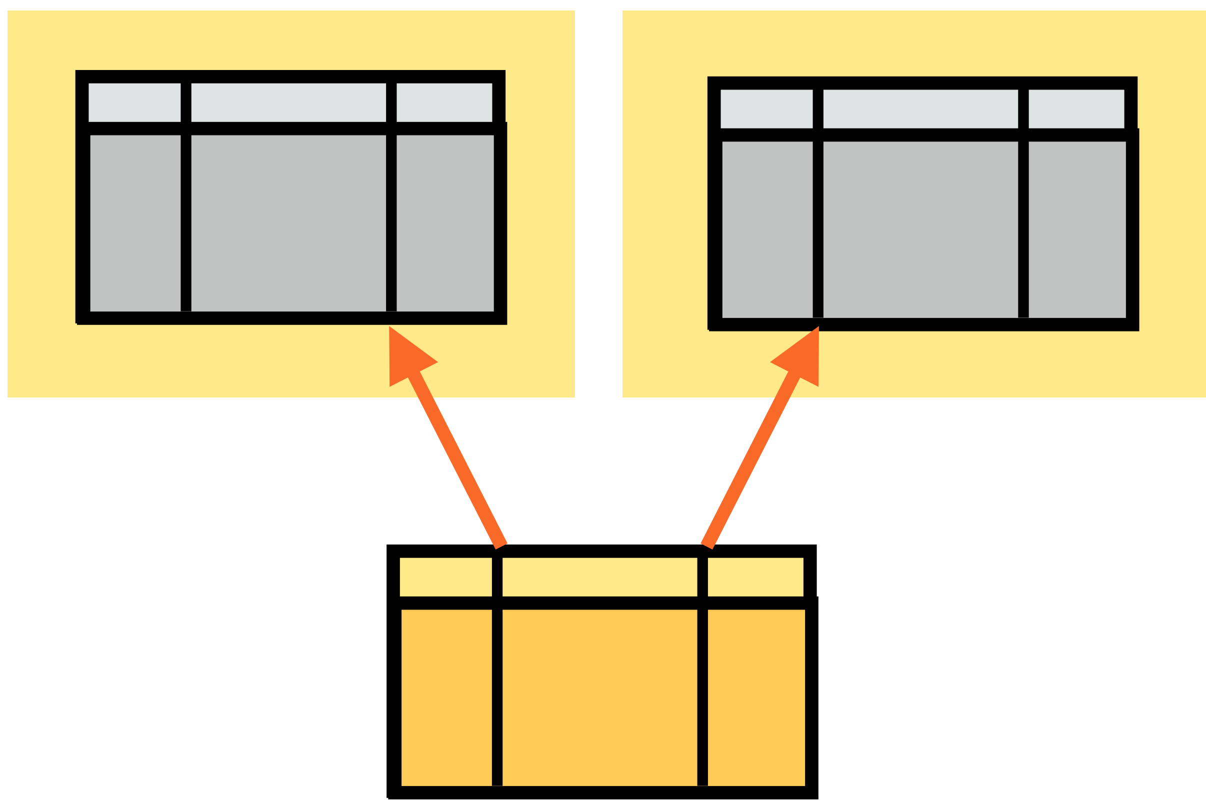
analog hardware is not configured yet

{ Goals, Blocks, Wires, Used Blocks, Config }

Goals	Blocks	Wires	Used Blocks	Config
$E = 6800 - ES$	DAC[0]	$DAC[0].O$ — $MM[0].A$	$\varnothing$	$\varnothing$
$S = 4400 - ES$	ADC[0]	$DAC[0].O$ — $MM[0].X0$		
$\dot{ES} = 10^{-4}E \cdot S - 10^{-2}ES$	MM[0]	$MM[0].Z$ — $ADC[0].X$		
$ES(0) = 0$	DAC[1]	$MM[0].X$ — $ADC[0].X$		

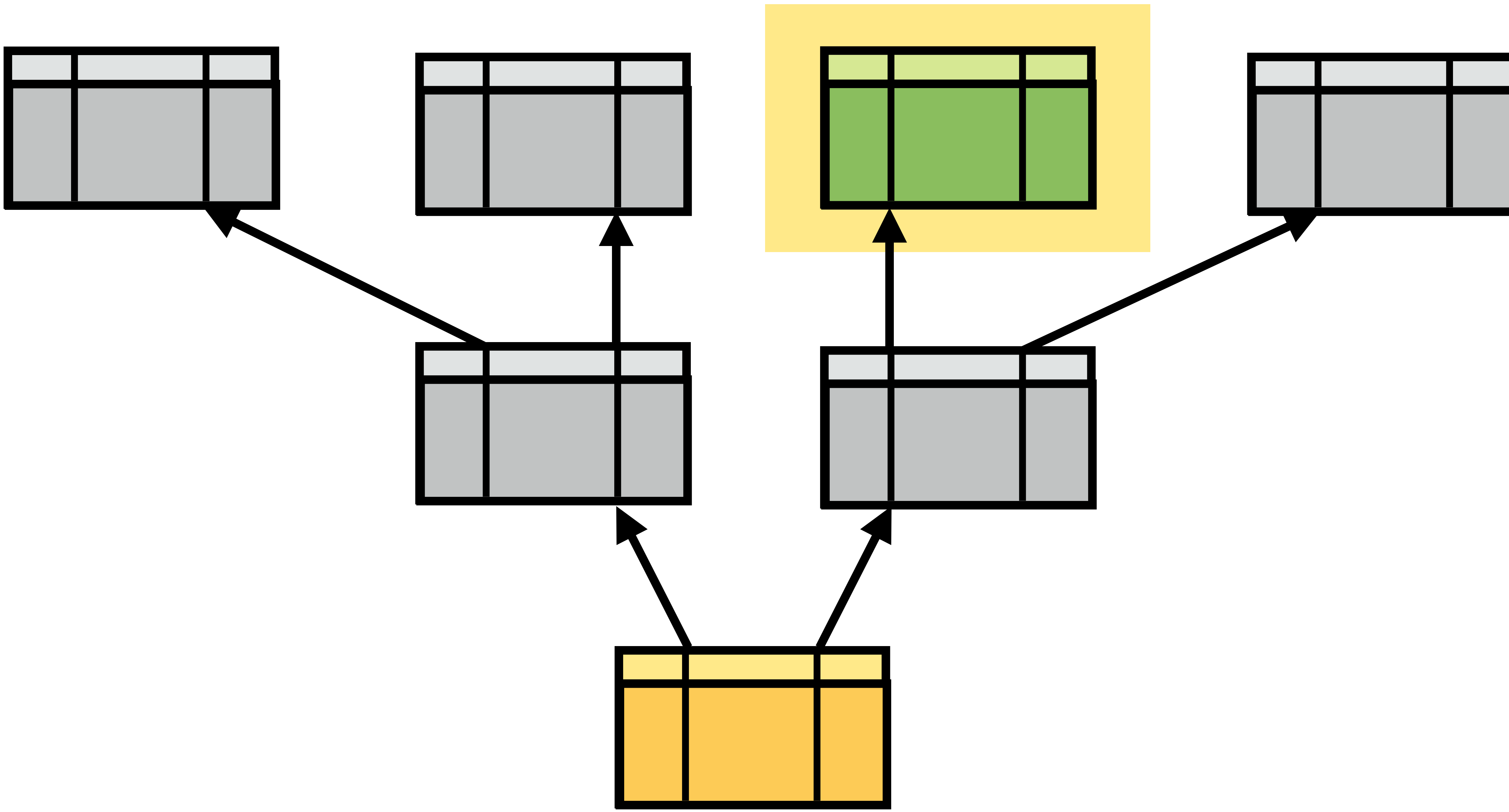
ARCO COMPILER OVERVIEW

new tableaus derived using **transition rules**



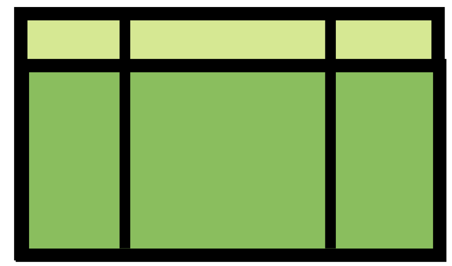
ARCO COMPILER OVERVIEW

Arco searches until a **solved tableau** is found



ARCO COMPILER

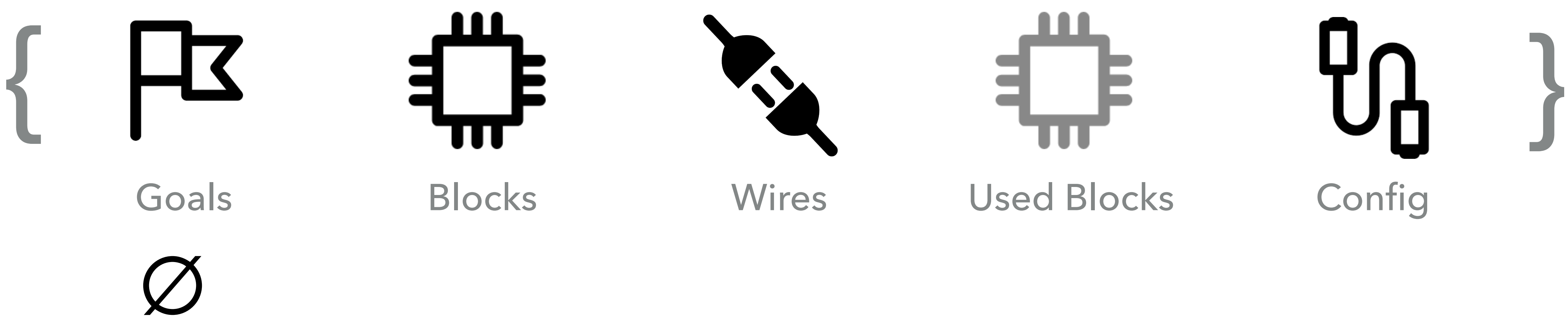
solved tableau : the final state of the search



ARCO COMPILER

solved tableau : the final state of the search

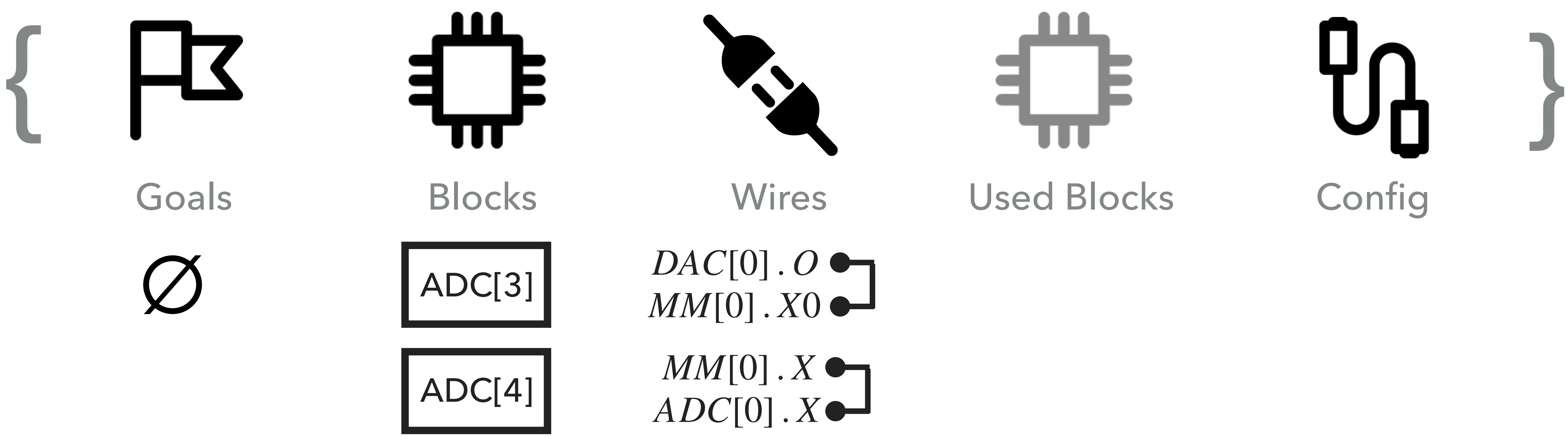
no goals left



ARCO COMPILER

solved tableau : the final state of the search

remaining blocks, wires



ARCO COMPILER

solved tableau : the final state of the search

blocks in use

Goals | Blocks | Wires | Used Blocks | Config

Goals: $\varnothing$

Blocks: ADC[3], ADC[4]

Wires:
$DAC[0].O$ — $MM[0].X0$
$MM[0].X$ — $ADC[0].X$

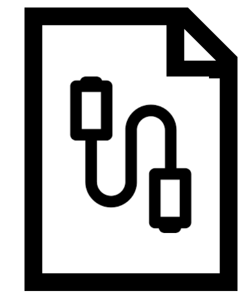
Used Blocks: MM[0], ADC[0], DAC[0], ADC[1]

Config:

ARCO COMPILER

solved tableau : the final state of the search

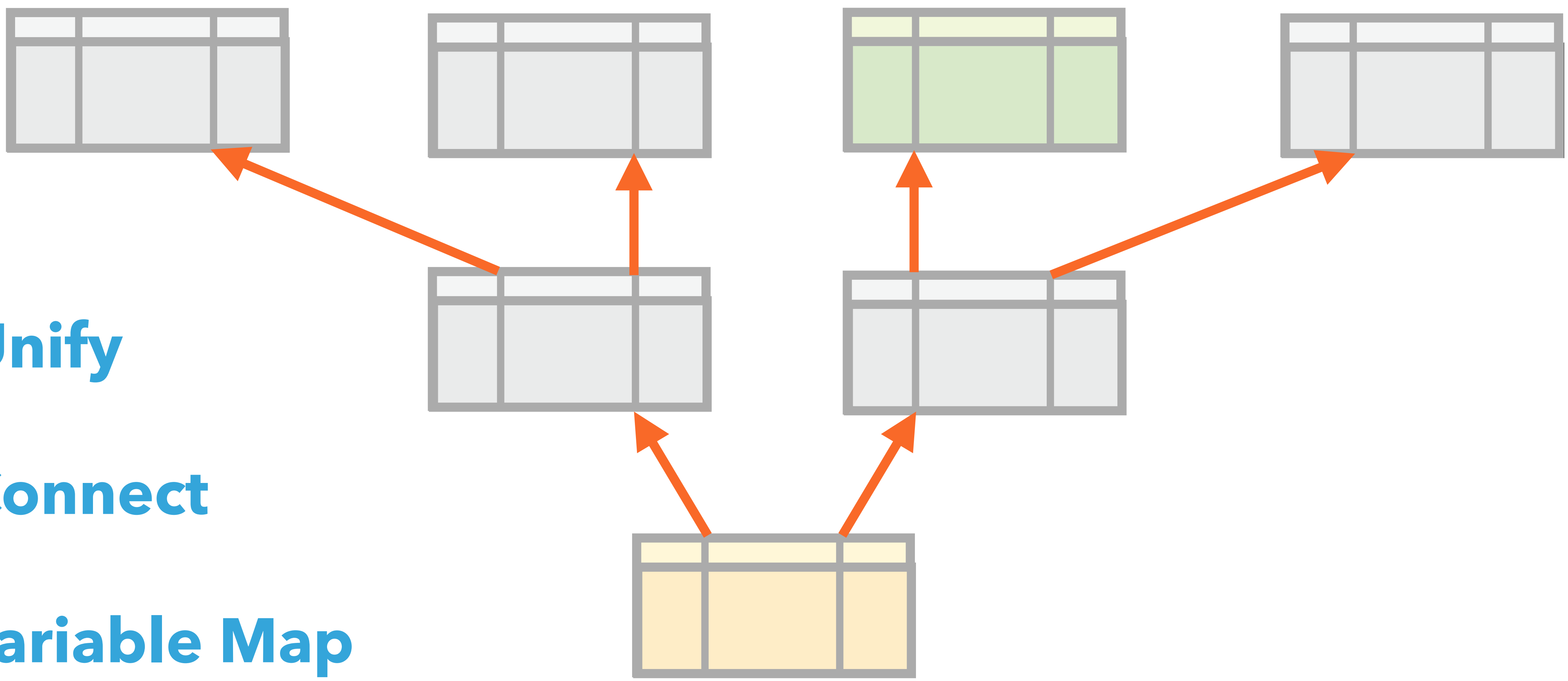
analog device configuration



Goals	Blocks	Wires	Used Blocks	Config
∅	ADC[3]	$DAC[0].O$ — $MM[0].X0$	MM[0]	**conn** DAC[0].Z **to** MM[0].A
	ADC[4]	$MM[0].X$ — $ADC[0].X$	ADC[0]	**conn** MM[0].Z **to** ADC[0].X
			DAC[0]	**set** DAC[0].X = 10^{-4}
			ADC[1]	**lbl** ADC[2].O = “ES”

ARCO COMPILER OVERVIEW

Arco derives new tableaus using **transition rules**



Unify

Connect

Variable Map

ARCO COMPILER OVERVIEW

Arco derives new tableaus using **transition rules**

Unify

Connect

Variable Map

UNIFICATION TRANSITION

$$E = 6800 - ES$$

{ Goals | Blocks | Wires | Used Blocks | Config }

Goals

$$S = 4400 - ES$$

$$\dot{ES} = 10^{-4}E \cdot S - 10^{-2}ES$$

$$ES(0) = 0$$

$$E = 6800 - ES$$

Blocks

DAC[0]

ADC[0]

MM[0]

DAC[1]

Wires

$DAC[0].O$ — $MM[0].A$

$DAC[0].O$ — $MM[0].X0$

$MM[0].Z$ — $ADC[0].X$

$MM[0].X$ — $ADC[0].X$

UNIFICATION TRANSITION

Goals

Blocks

$$S = 4400 - ES$$

$$\dot{E}S = 10^{-4}E \cdot S - 10^{-2}ES$$

$$ES(0) = 0$$

$$E = 6800 - ES$$

MM[0]

MM[0]

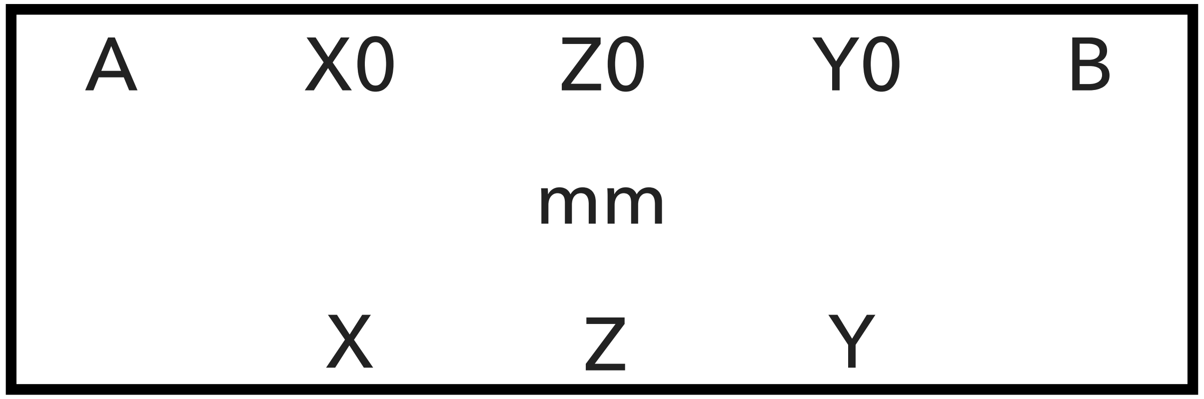
Goals

$$S = 4400 - ES$$

$$\dot{ES} = 10^{-4}E \cdot S - 10^{-2}ES$$

$$ES(0) = 0$$

$$E = 6800 - ES$$



$$E = 6800 - ES$$

Goals

$$S = 4400 - ES$$

$$\dot{ES} = 10^{-4}E \cdot S - 10^{-2}ES$$

$$ES(0) = 0$$

$$E = 6800 - ES$$

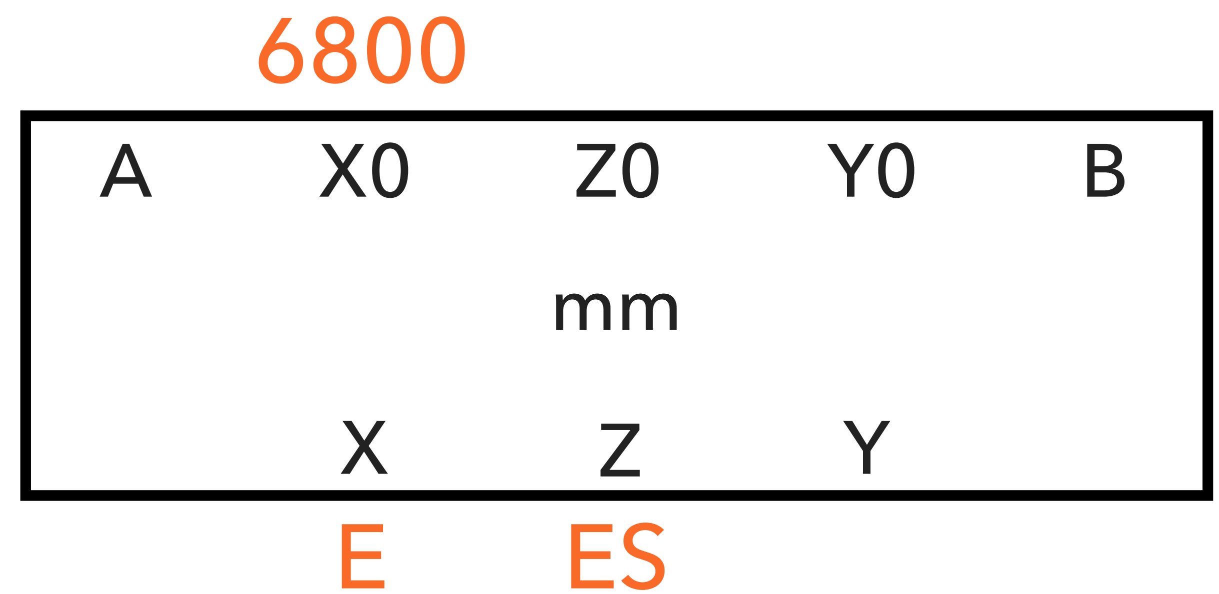
MM[0]

A	X0	Z0	Y0	B
		mm		
	X	Z	Y	

$$E = 6800 - ES$$

$$X = X0 - Z$$

MM[0]



Goals

$$S = 4400 - ES$$
$$\dot{E}S = 10^{-4}E \cdot S - 10^{-2}ES$$
$$ES(0) = 0$$

$$\cancel{E = 6800 - ES}$$

$$MM[0].X0 = 6800$$

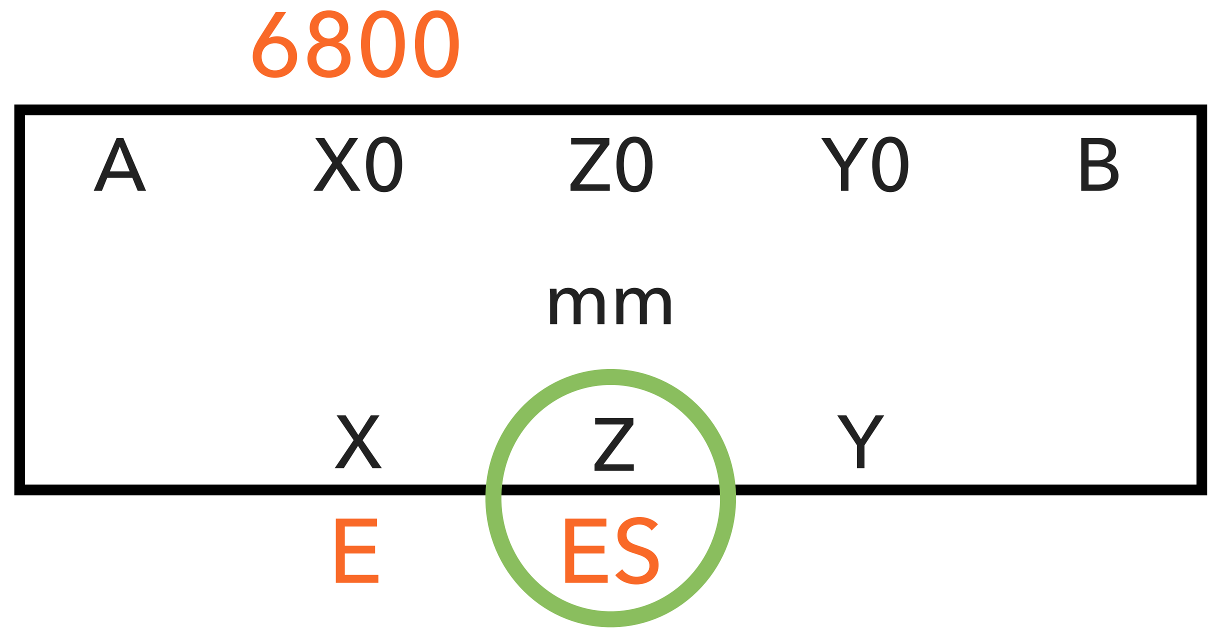
$$MM[0].X = E$$

$$MM[0].Z = ES$$

$$E = 6800 - ES$$

$$X = X0 - Z$$

MM[0]



Goals

$$S = 4400 - ES$$

$$\dot{ES} = 10^{-4}E \cdot S - 10^{-2}ES$$
$$ES(0) = 0$$

$$\cancel{E = 6800 - ES}$$

$$MM[0]\,.\,X0 = 6800$$

$$MM[0]\,.\,X = E$$

$$MM[0]\,.\,Z = ES$$

$$\dot{ES} = 10^{-4}E \cdot S - 10^{-2}ES$$
$$ES(0) = 0$$

$$\dot{Z} = A \cdot X \cdot Y - B \cdot Z$$
$$Z(0) = Z0$$

Goals

$S = 4400 - ES$

$\dot{ES} = 10^{-4}E \cdot S - 10^{-2}ES$

$ES(0) = 0$

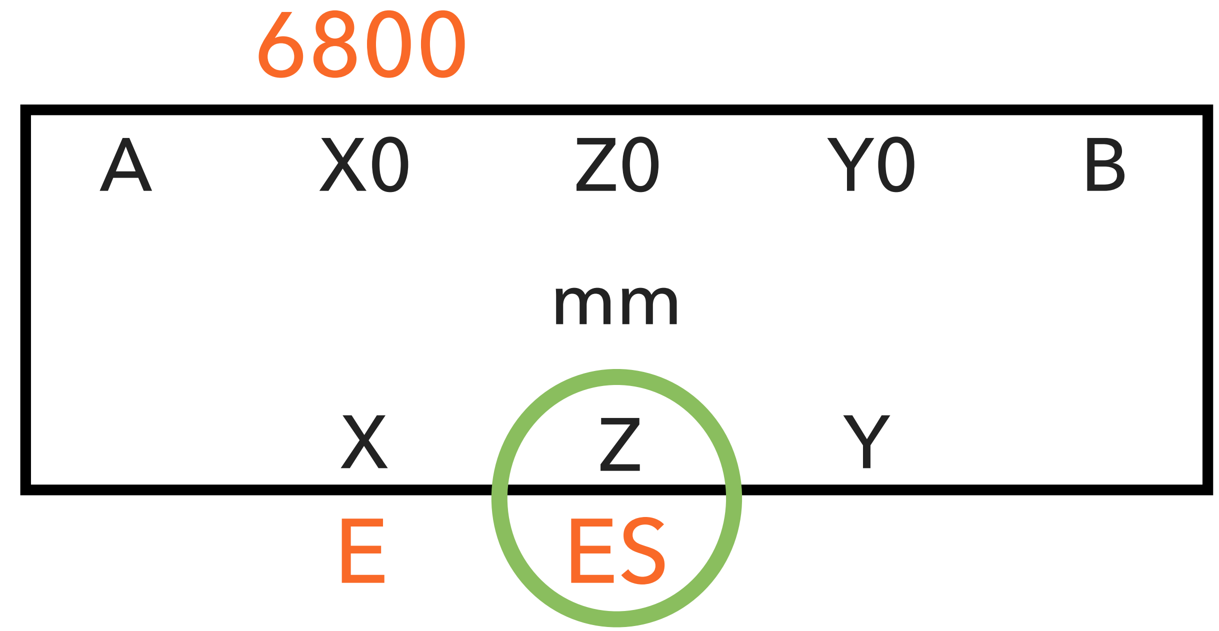
$\cancel{E = 6800 - ES}$

$MM[0].X0 = 6800$

$MM[0].X = E$

$MM[0].Z = ES$

MM[0]



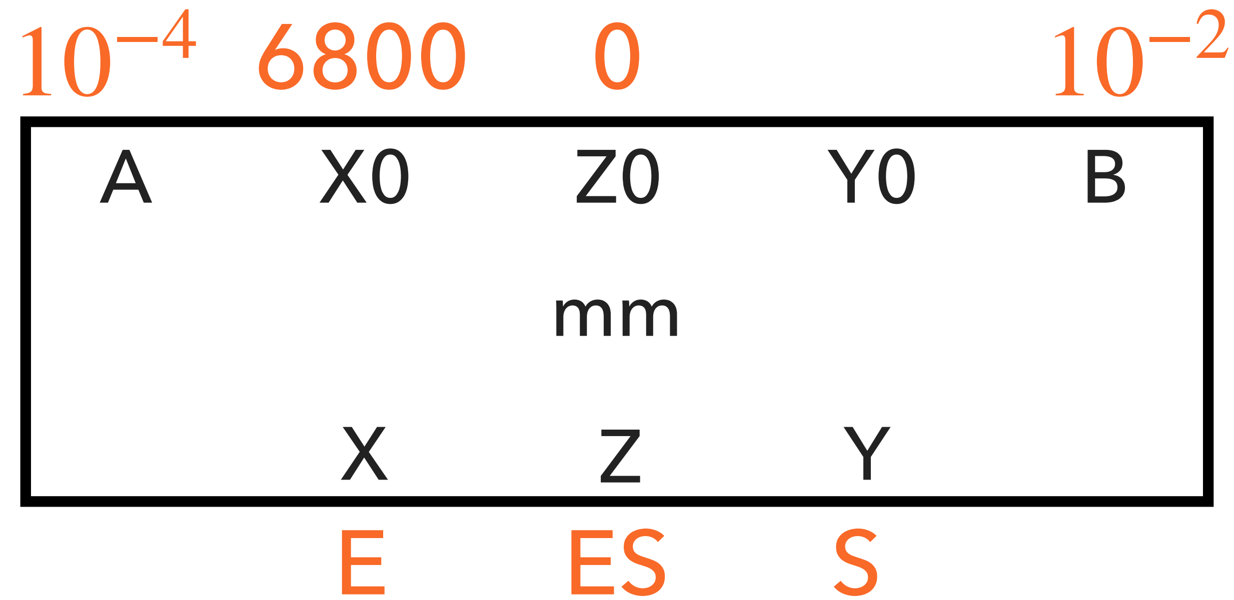
$$\dot{ES} = 10^{-4}E \cdot S - 10^{-2}ES$$

$$ES(0) = 0$$

$$\dot{ES} = A \cdot E \cdot Y - B \cdot ES$$

$$ES(0) = Z0$$

MM[0]



Goals

$$S = 4400 - ES$$

$$\cancel{\dot{ES} = 10^{-4}E \cdot S - 10^{-2}ES}$$

$$\cancel{ES(0) = 0}$$

$$\cancel{E = 6800 - ES}$$

$$MM[0].X0 = 6800$$

$$MM[0].X = E$$

$$MM[0].Z = ES$$

$$MM[0].A = 10^{-4}$$

$$MM[0].B = 10^{-2}$$

$$MM[0].Y = S$$

$$MM[0].Z0 = 0$$

$$\dot{ES} = 10^{-4}E \cdot S - 10^{-2}ES$$

$$ES(0) = 0$$

$$\dot{ES} = A \cdot E \cdot Y - B \cdot ES$$

$$ES(0) = Z0$$

Goals

$$S = 4400 - ES$$

$$\cancel{\dot{ES} = 10^{-4} E \cdot S - 10^{-2} ES}$$

$$\cancel{ES(0) = 0}$$

$$\cancel{E = 6800 - ES}$$

$$MM[0]\,.\,X0 = 6800$$

$$MM[0]\,.\,X = E$$

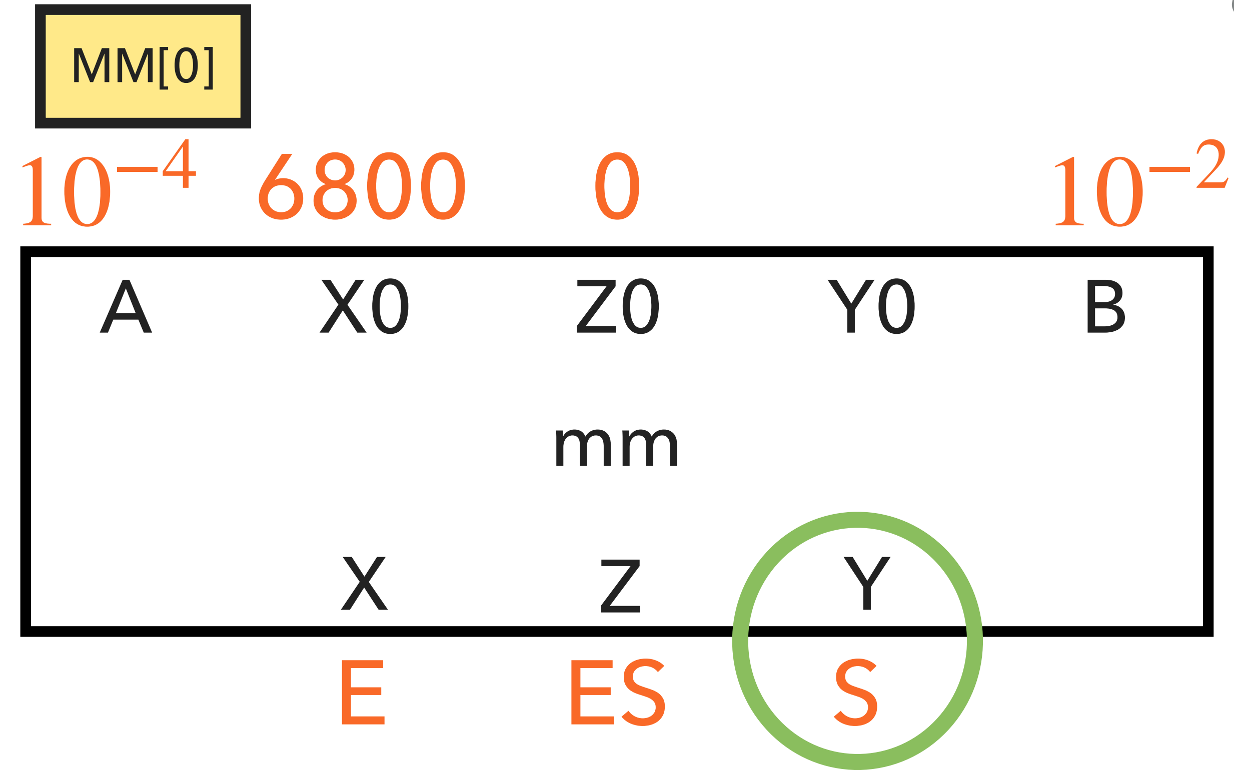
$$MM[0]\,.\,Z = ES$$

$$MM[0]\,.\,A = 10^{-4}$$

$$MM[0]\,.\,B = 10^{-2}$$

$$MM[0]\,.\,Y = S$$

$$MM[0]\,.\,Z0 = 0$$



$$S = 4400 - ES$$

$$S = Y0 - ES$$

Goals

$\cancel{S = 4400 - ES}$

$\cancel{\dot{ES} = 10^{-4}E \cdot S - 10^{-2}ES}$

$\cancel{ES(0) = 0}$

$\cancel{E = 6800 - ES}$

$MM[0]\,.\,X0 = 6800$

$MM[0]\,.\,X = E$

$MM[0]\,.\,Z = ES$

$MM[0]\,.\,A = 10^{-4}$

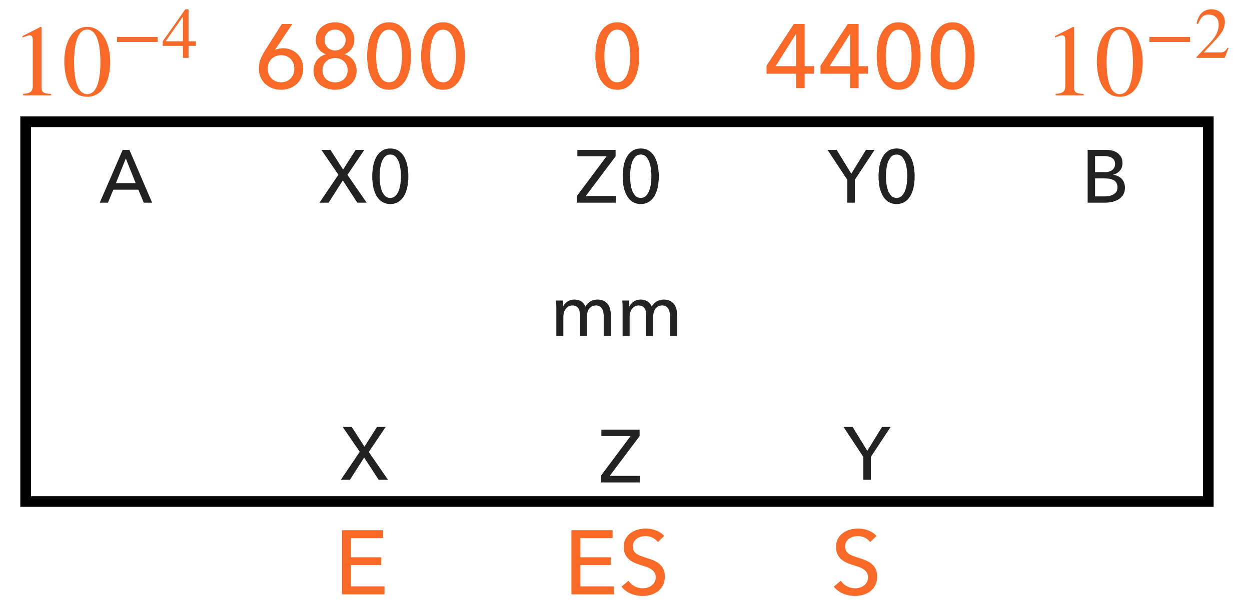
$MM[0]\,.\,B = 10^{-2}$

$MM[0]\,.\,Y = S$

$MM[0]\,.\,Z0 = 0$

$MM[0]\,.\,Y0 = 4400$

MM[0]



$$S = 4400 - ES$$

$$S = Y0 - ES$$

Goals

$\cancel{S = 4400 - ES}$

$\cancel{\dot{ES} = 10^{-4} E \cdot S - 10^{-2} ES}$

$\cancel{ES(0) = 0}$

$\cancel{E = 6800 - ES}$

$MM[0]\,.\,X0 = 6800$

$MM[0]\,.\,X = E$

$MM[0]\,.\,Z = ES$

$MM[0]\,.\,A = 10^{-4}$

$MM[0]\,.\,B = 10^{-2}$

$MM[0]\,.\,Y = S$

$MM[0]\,.\,Z0 = 0$

$MM[0]\,.\,Y0 = 4400$

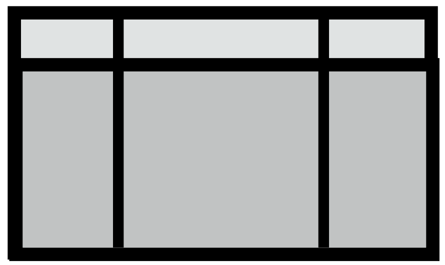
MM[0]

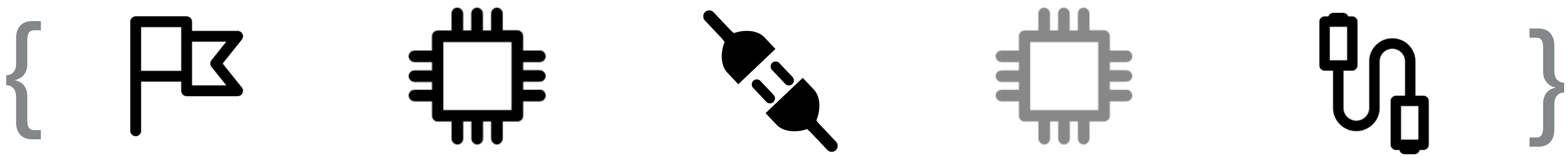
10^{-4}	6800	0	4400	10^{-2}
A	X0	Z0	Y0	B
		mm		
	X	Z	Y	
	E	ES	S	

$$S = 4400 - ES$$

$$S = Y0 - ES$$

UNIFICATION TRANSITION





$$S = 4400 - ES$$
$$\dot{E}S = 10^{-4}E \cdot S - 10^{-2}ES$$
$$ES(0) = 0$$
$$E = 6800 - ES$$

DAC[0]

ADC[0]

MM[0]

DAC[1]

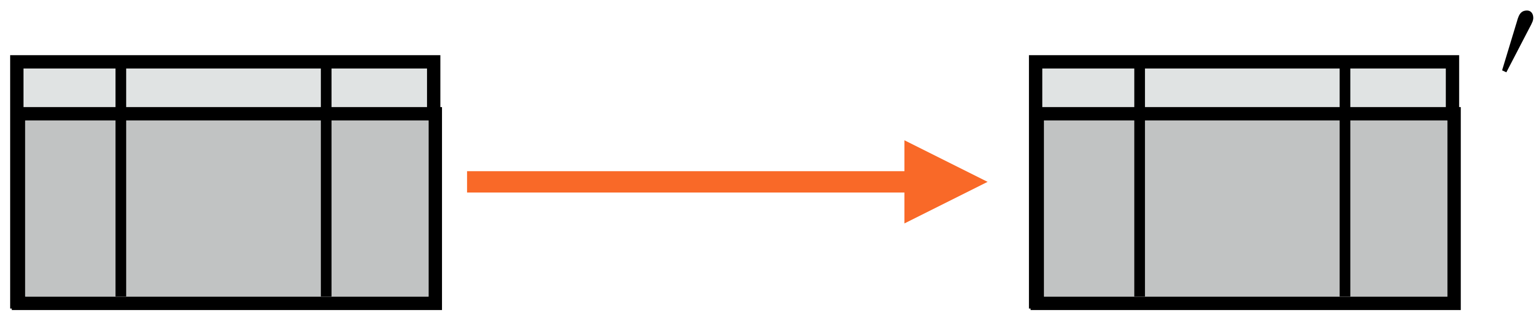
$DAC[0].O$ — $MM[0].A$

$DAC[0].O$ — $MM[0].X0$

$MM[0].Z$ — $ADC[0].X$

$MM[0].X$ — $ADC[0].X$

UNIFICATION TRANSITION



Goals

$\cancel{S = 4400 - ES}$

$\cancel{\dot{ES} = 10^{-4}E \cdot S - 10^{-2}ES}$

$\cancel{ES(0) = 0}$

$\cancel{E = 6800 - ES}$

$MM[0] . X0 = 6800$

$MM[0] . X = E$

$MM[0] . Z = ES$

Blocks

$\boxed{\text{DAC[0]}}$

$\boxed{\text{ADC[0]}}$

$\cancel{\boxed{\text{MM[0]}}}$

$\boxed{\text{DAC[1]}}$

Wires

$DAC[0] . O$ — $MM[0] . A$

$DAC[0] . O$ — $MM[0] . X0$

$MM[0] . Z$ — $ADC[0] . X$

$MM[0] . X$ — $ADC[0] . X$

Used Blocks

$\boxed{\text{MM[0]}}$

ARCO COMPILER OVERVIEW

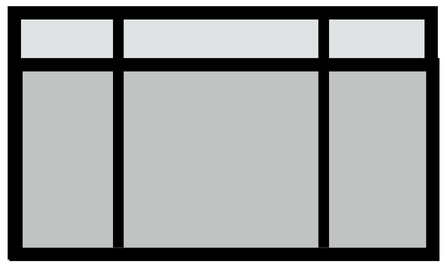
Arco derives new tableaus using **transition rules**

Unify

Connect

Variable Map

CONNECTION TRANSITION



Goals

$MM[0].A = DAC[0].O$

$DAC[0].X = 10^{-4}$

$ADC[0].X = MM[0].Z$

$ADC[0].O = ES$

$MM[0].B = DAC[0].O$

$DAC[0].X = 10^{-2}$

$ADC[1].X = MM[0].X$

$ADC[1].O = E$

Blocks

MM[1]

MM[2]

ADC[3]

ADC[4]

Wires

$DAC[0].O$ – $MM[0].A$

$DAC[0].O$ – $MM[0].X0$

$MM[0].Z$ – $ADC[0].X$

$MM[0].X$ – $ADC[0].X$

Used Blocks

MM[0]

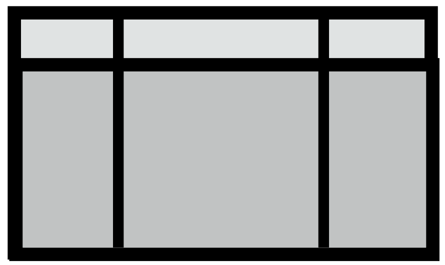
DAC[0]

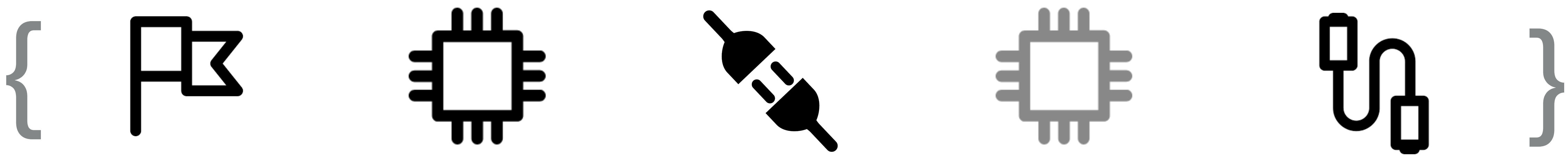
DAC[1]

ADC[0]

ADC[1]

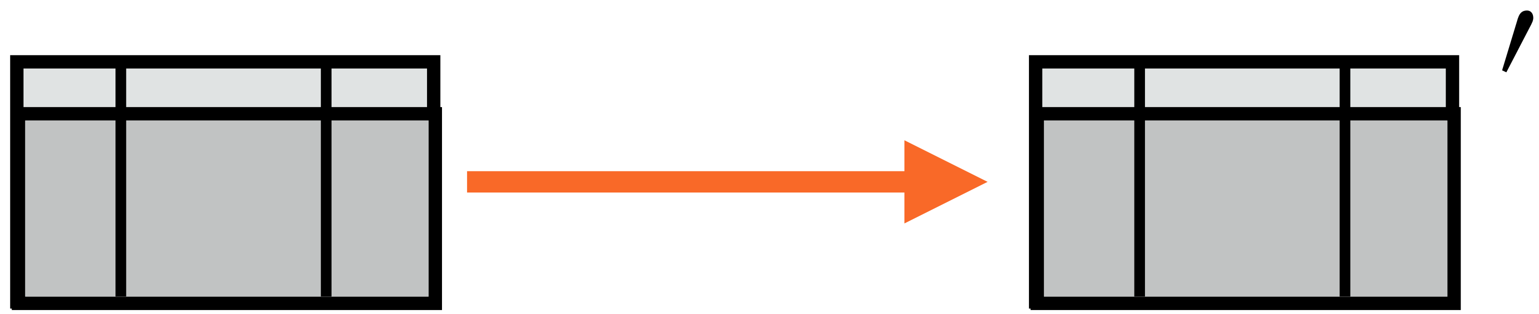
CONNECTION TRANSITION





Goals	Blocks	Wires	Used Blocks	Config
$MM[0].A = DAC[0].O$	MM[1]	$DAC[0].O$ $MM[0].A$	MM[0]	
$DAC[0].X = 10^{-4}$	MM[2]	$DAC[0].O$ $MM[0].X0$	DAC[0]	
$ADC[0].X = MM[0].Z$	ADC[3]	$MM[0].Z$ $ADC[0].X$	DAC[1]	
$ADC[0].O = ES$	ADC[4]	$MM[0].X$ $ADC[0].X$	ADC[0]	
$MM[0].B = DAC[0].O$			ADC[1]	
$DAC[0].X = 10^{-2}$				
$ADC[1].X = MM[0].X$				
$ADC[1].O = E$				

CONNECTION TRANSITION



Goals

$\cancel{MM[0].A = DAC[0].O}$

$DAC[0].X = 10^{-4}$

$ADC[0].X = MM[0].Z$

$ADC[0].O = ES$

$MM[0].B = DAC[0].O$

$DAC[0].X = 10^{-2}$

$ADC[1].X = MM[0].X$

$ADC[1].O = E$

Blocks

MM[1]

MM[2]

ADC[3]

ADC[4]

Wires

$\cancel{DAC[0].O}$ — $\cancel{MM[0].A}$

$DAC[0].O$ — $MM[0].X0$

$MM[0].Z$ — $ADC[0].X$

$MM[0].X$ — $ADC[0].X$

Used Blocks

MM[0]

DAC[0]

DAC[1]

ADC[0]

ADC[1]

Config

conn DAC[0].Z **to** MM[0].A

ARCO COMPILER OVERVIEW

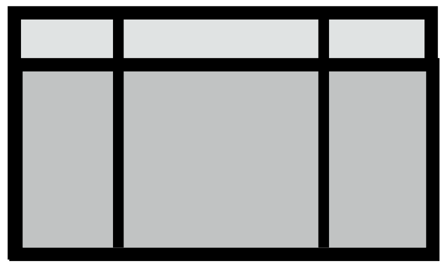
Arco derives new tableaus using **transition rules**

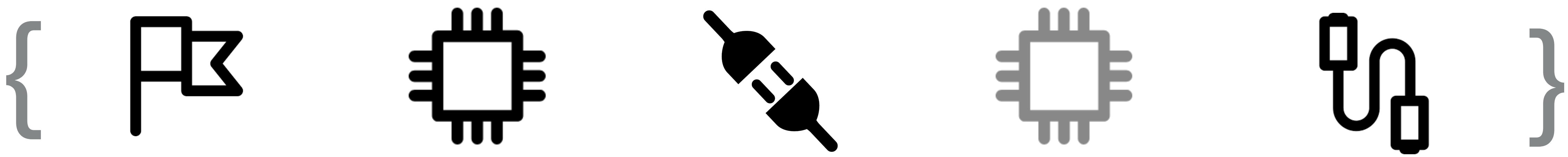
Unify

Connect

Variable Map

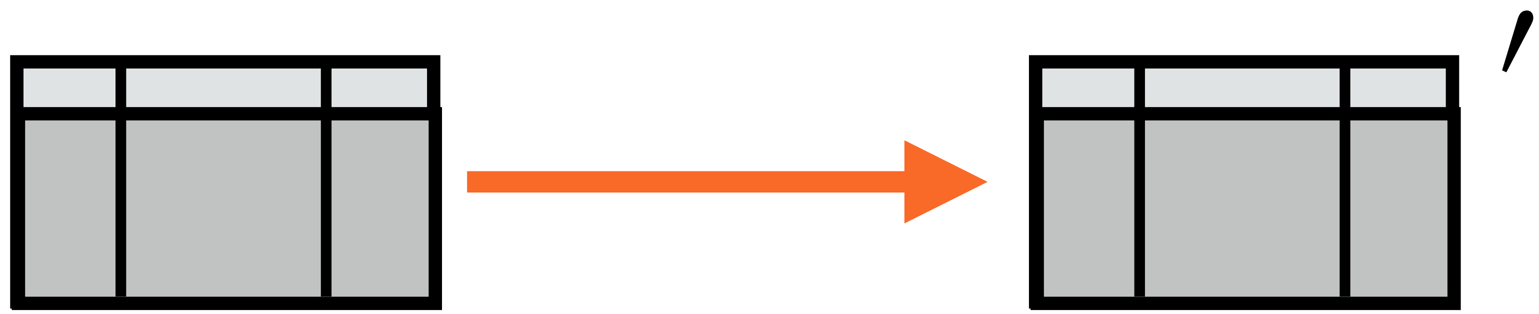
VARIABLE/VALUE MAPPING TRANSITION





Goals	Blocks	Wires	Used Blocks	Config
$MM[0].A = DAC[0].O$	MM[1]	$DAC[0].O$ — $MM[0].X0$	MM[0]	**conn** DAC[0].Z **to** MM[0].A
$DAC[0].X = 10^{-4}$	MM[2]	$MM[0].Z$ — $ADC[0].X$	DAC[0]	
$ADC[0].X = MM[0].Z$	ADC[3]	$MM[0].X$ — $ADC[0].X$	DAC[1]	
$ADC[0].O = ES$	ADC[4]	$MM[0].Y$ — $ADC[0].X$	ADC[0]	
$MM[0].B = DAC[0].O$			ADC[1]	
$DAC[0].X = 10^{-2}$				
$ADC[1].X = MM[0].X$				
$ADC[1].O = E$				

VARIABLE/VALUE MAPPING TRANSITION



Goals

- $MM[0].A = DAC[0].O$
- $\cancel{DAC[0].X = 10^{-4}}$
- $ADC[0].X = MM[0].Z$
- $ADC[0].O = ES$
- $MM[0].B = DAC[0].O$
- $DAC[0].X = 10^{-2}$
- $ADC[1].X = MM[0].X$
- $ADC[1].O = E$

Blocks

- MM[1]
- MM[2]
- ADC[3]
- ADC[4]

Wires

- $DAC[0].O$ – $MM[0].X0$
- $MM[0].Z$ – $ADC[0].X$
- $MM[0].X$ – $ADC[0].X$
- $MM[0].Y$ – $ADC[0].X$

Used Blocks

- MM[0]
- DAC[0]
- DAC[1]
- ADC[0]
- ADC[1]

Config

conn DAC[0].Z **to** MM[0].A

set DAC[0].X = 10^{-4}

VARIABLE/VALUE MAPPING TRANSITION

{ Goals, Blocks, Wires, Used Blocks, Config }

Goals

$MM[0].A = DAC[0].O$

$\cancel{DAC[0].X = 10^{-4}}$

$ADC[0].X = MM[0].Z$

$ADC[0].O = ES$

$MM[0].B = DAC[0].O$

$DAC[0].X = 10^{-2}$

$ADC[1].X = MM[0].X$

$ADC[1].O = E$

Blocks

MM[1]

MM[2]

ADC[3]

ADC[4]

Wires

$DAC[0].O$ – $MM[0].X0$

$MM[0].Z$ – $ADC[0].X$

$MM[0].X$ – $ADC[0].X$

$MM[0].Y$ – $ADC[0].X$

Used Blocks

MM[0]

DAC[0]

DAC[1]

ADC[0]

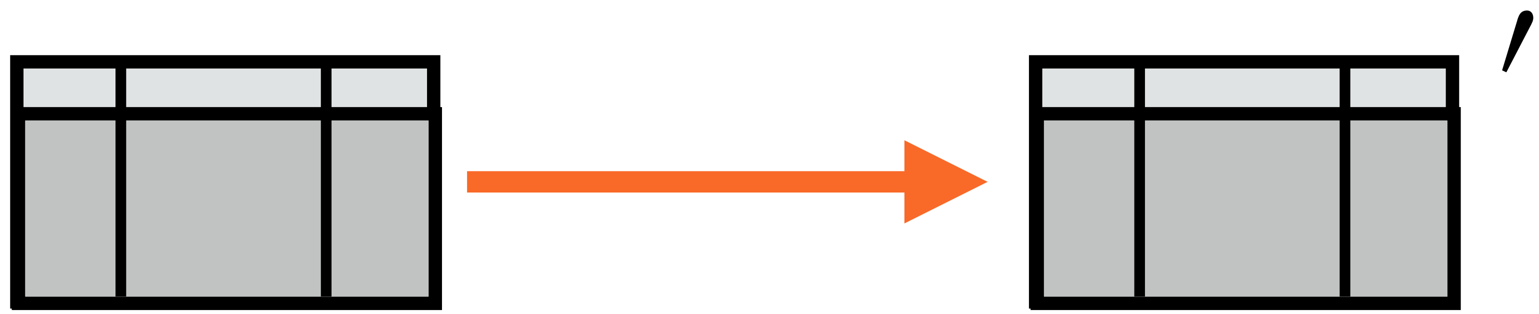
ADC[1]

Config

conn DAC[0].Z **to** MM[0].A

set DAC[0].X = 10^{-4}

VARIABLE/VALUE MAPPING TRANSITION



{ Goals, Blocks, Wires, Used Blocks, Config }

Goals

$MM[0].A = DAC[0].O$

$\cancel{DAC[0].X = 10^{-4}}$

$ADC[0].X = MM[0].Z$

$\cancel{ADC[0].O = ES}$

$MM[0].B = DAC[0].O$

$DAC[0].X = 10^{-2}$

$ADC[1].X = MM[0].X$

$ADC[1].O = E$

Blocks

MM[1]

MM[2]

ADC[3]

ADC[4]

Wires

$DAC[0].O$ — $MM[0].X0$

$MM[0].Z$ — $ADC[0].X$

$MM[0].X$ — $ADC[0].X$

$MM[0].Y$ — $ADC[0].X$

Used Blocks

MM[0]

DAC[0]

DAC[1]

ADC[0]

ADC[1]

Config

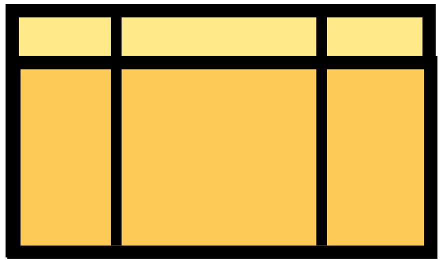
conn DAC[0].Z **to** MM[0].A

set DAC[0].X = 10^{-4}

lbl ADC[0].O = ES

ARCO COMPILER RECAP

Arco starts with an **initial tableau**



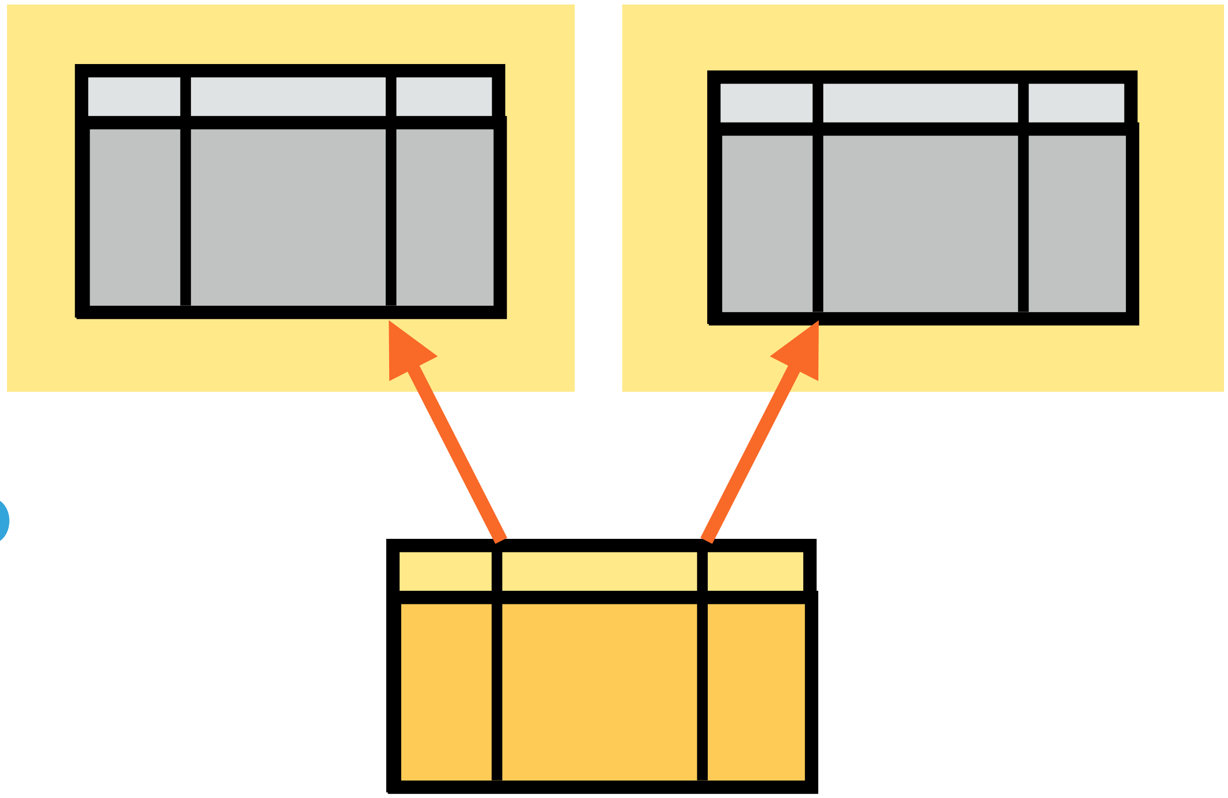
ARCO COMPILER RECAP

derives new tableaus using **transition rules**

Unify

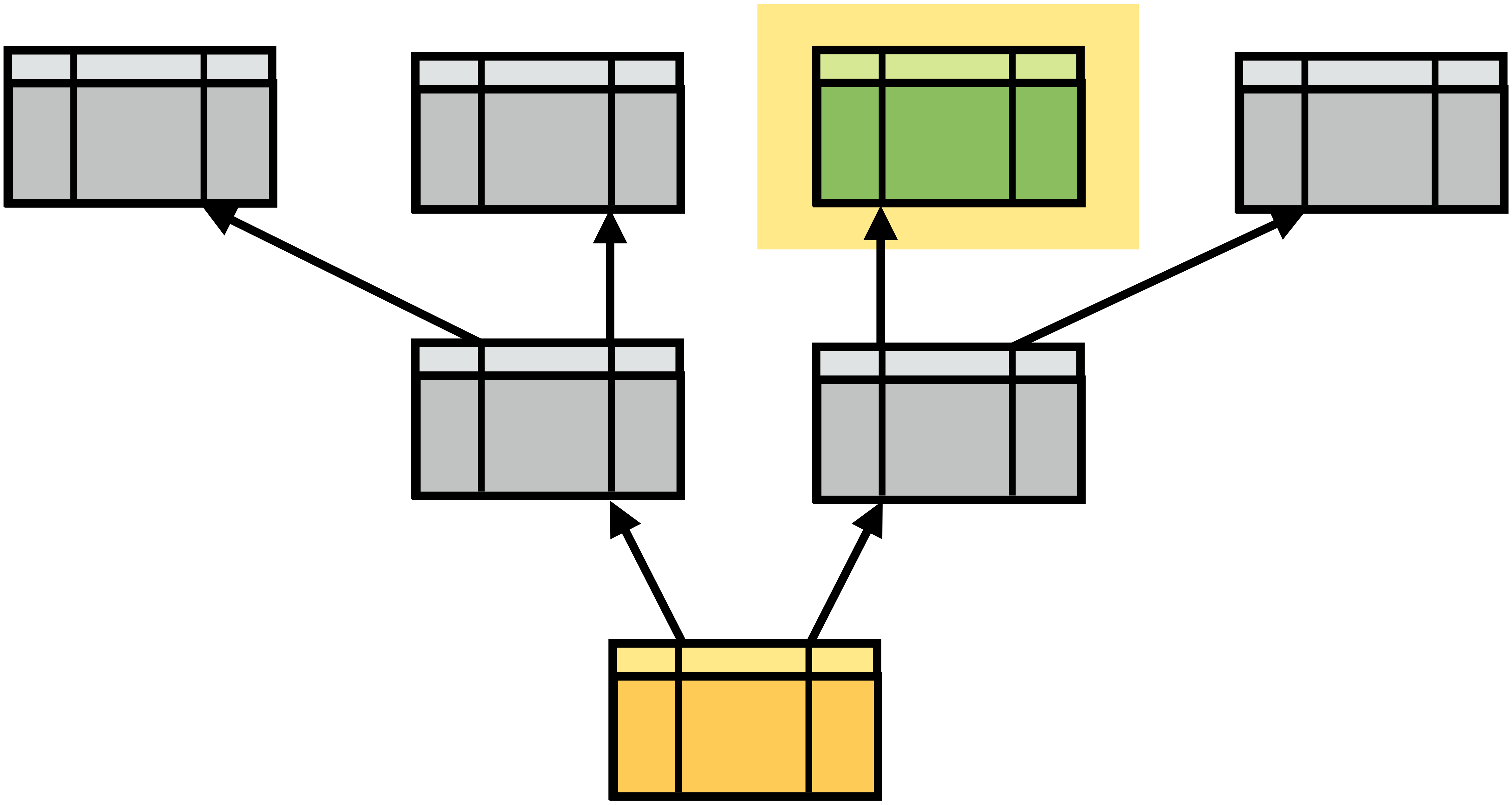
Connect

Variable Map



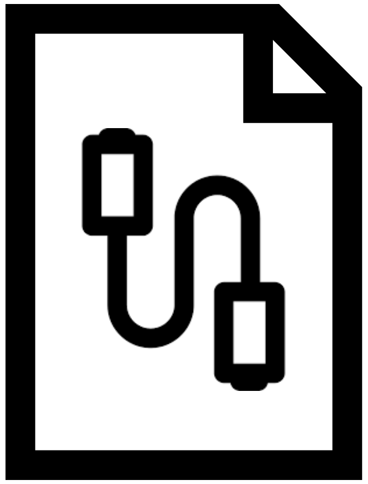
ARCO COMPILER RECAP

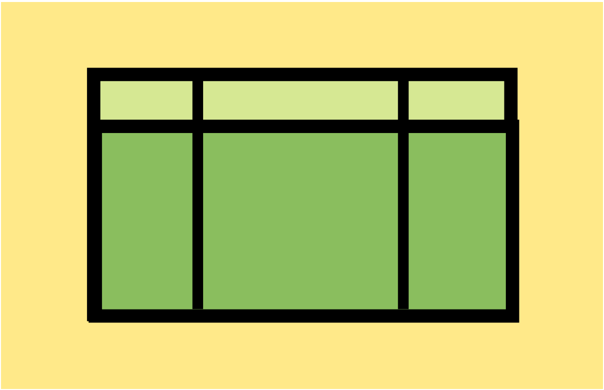
until a **solved tableau** is found



ARCO COMPILER RECAP

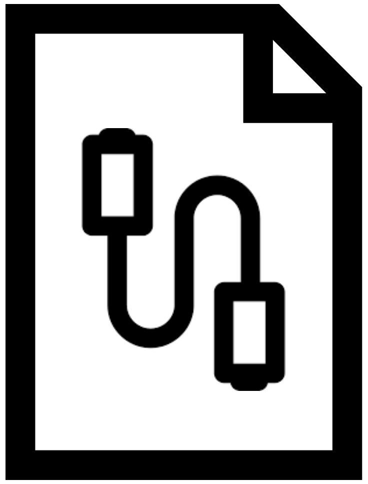
analog device configuration in solved tableau

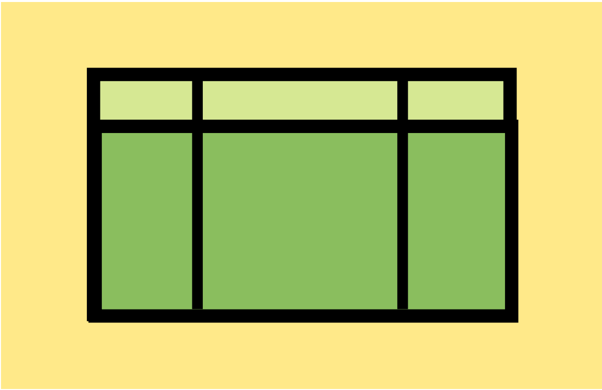




ARCO COMPILER RECAP

analog device configuration in solved tableau





ALGEBRAICALLY EQUIVALENT TO DYNAMICAL SYSTEM

creative use of available analog blocks to model dynamics

respects connectivity, block instance constraints

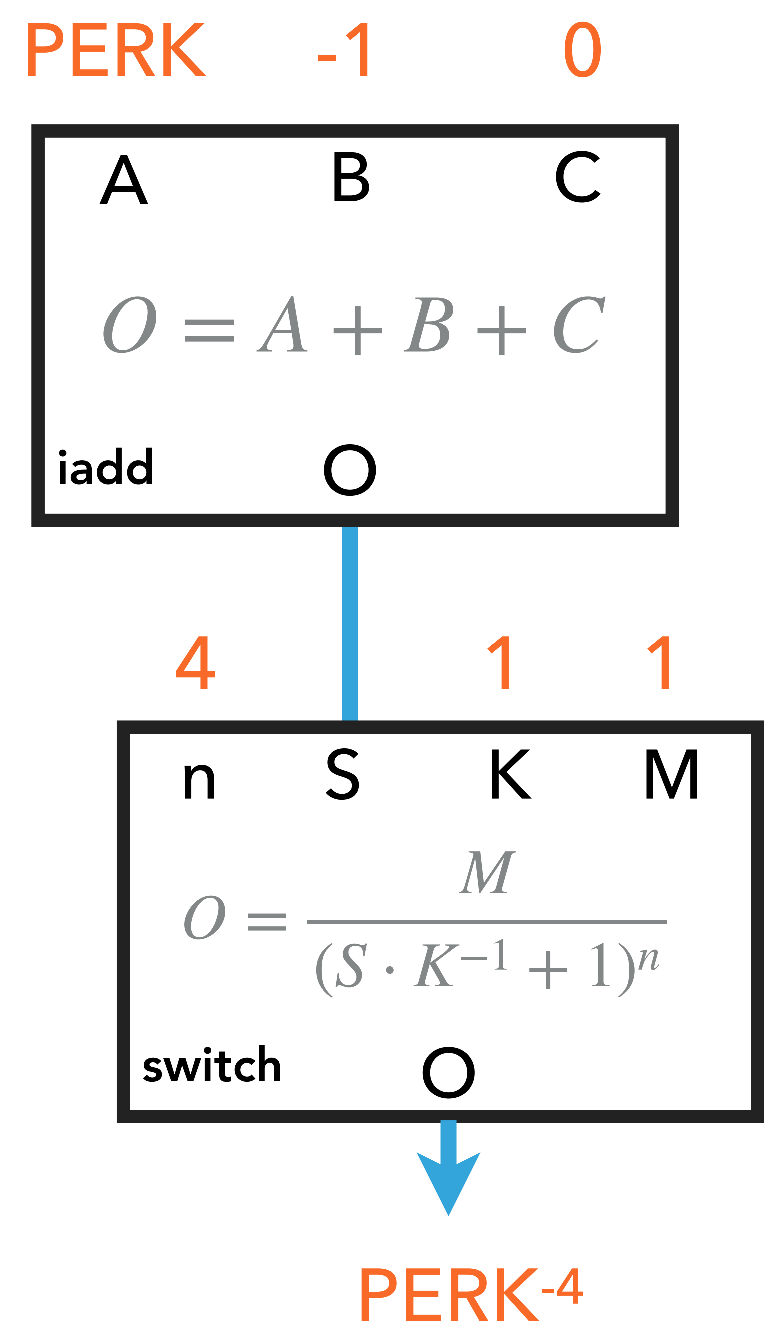
CASE STUDY 1: PERK-4

Task: model PERK^{-4} using analog hardware that does not directly support exponentiation.

$$O = \frac{M}{((A+B+C)\cdot K^{-1}+1)^n}$$

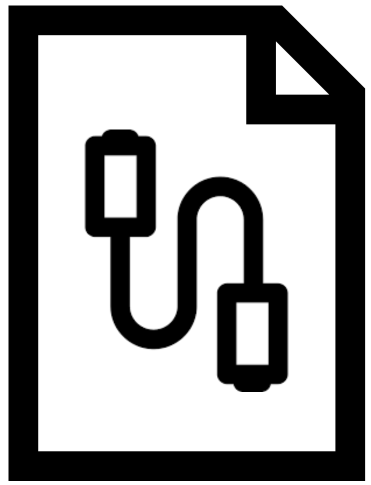
$$O = \frac{1}{((PERK+(-1)+0)\cdot 1^{-1}+1)^4}$$

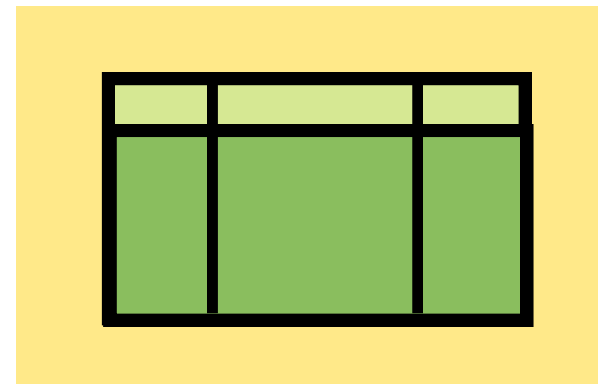
PERK^{-4}



ARCO COMPILER RECAP

analog device configuration in solved tableau



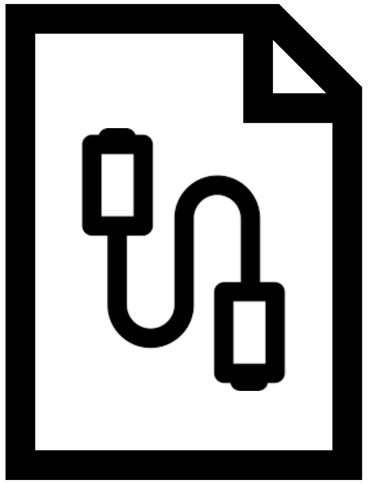


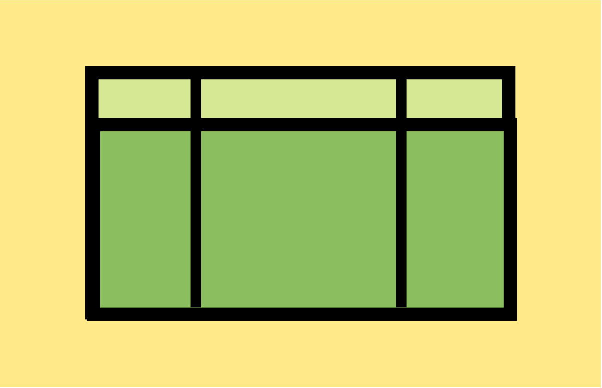
doesn't take into consideration

PHYSICAL LIMITATIONS OF HARDWARE

ARCO COMPILER RECAP

analog device configuration in solved tableau





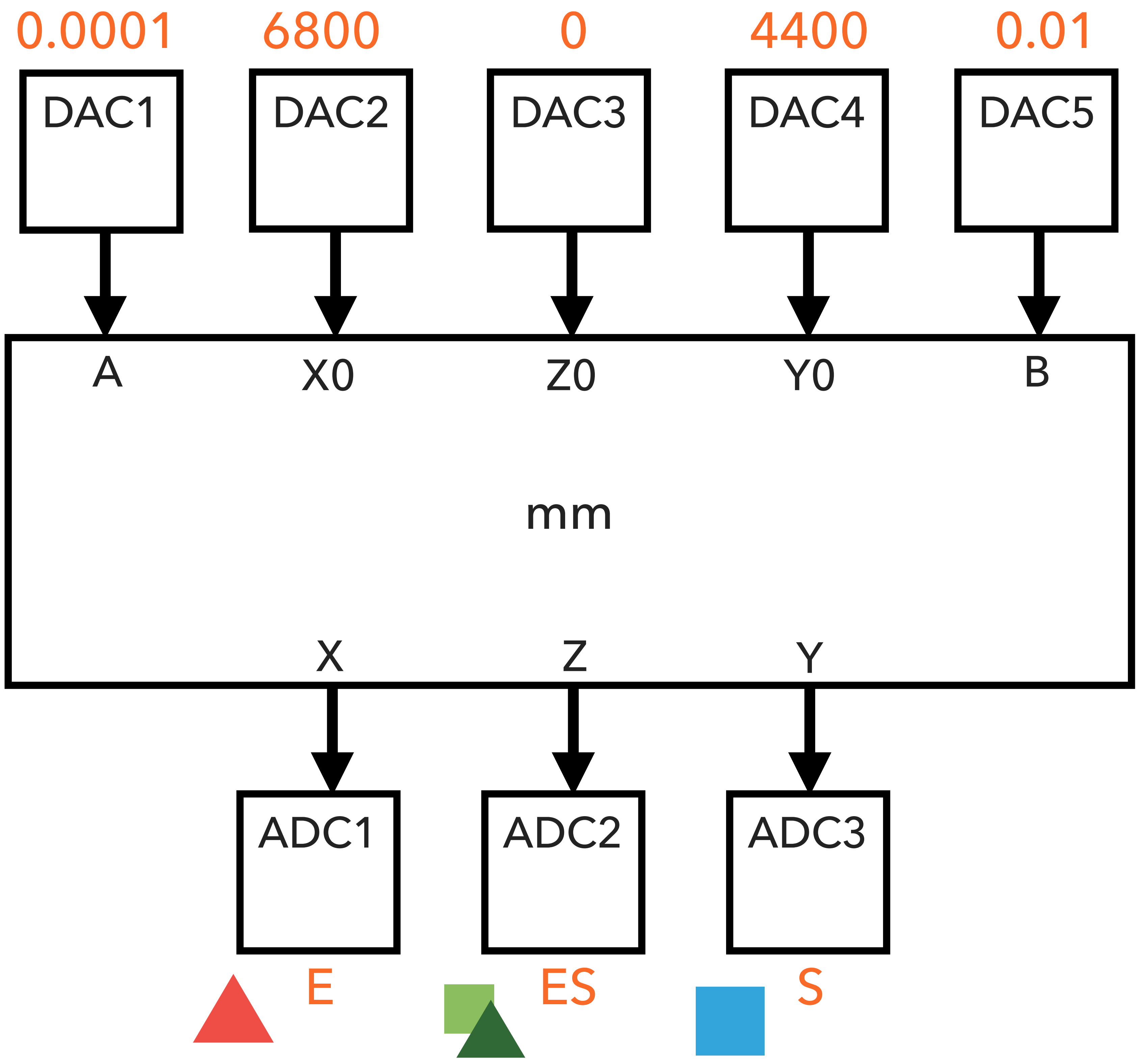
doesn't take into consideration

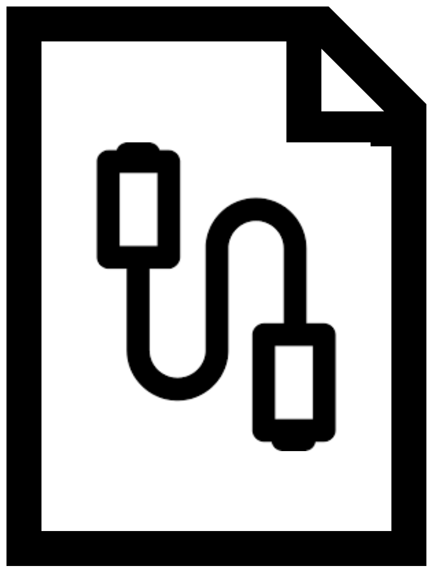
PHYSICAL LIMITATIONS OF HARDWARE

OPERATING RANGE CONSTRAINTS

SAMPLING RATES OF ADCS/DACS

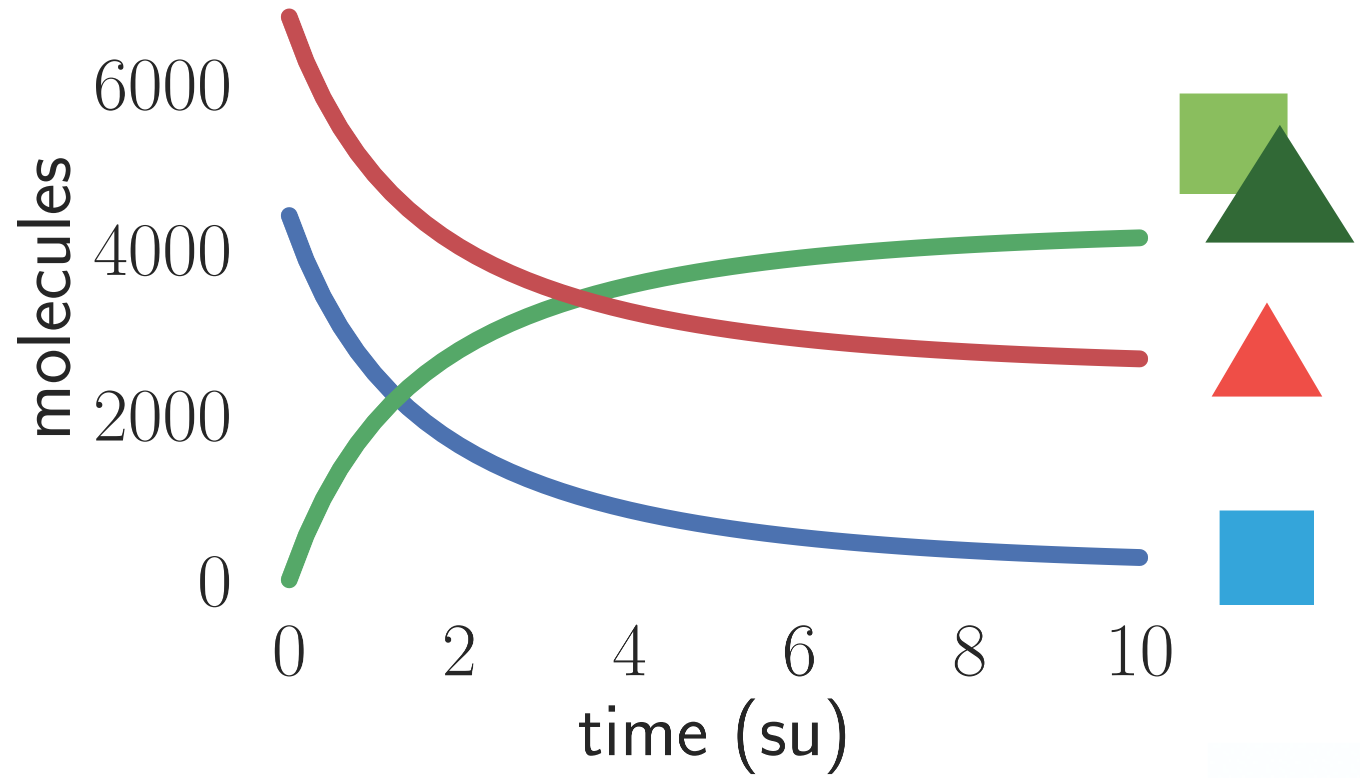
ANALOG DEVICE CONFIGURATION REVISITED

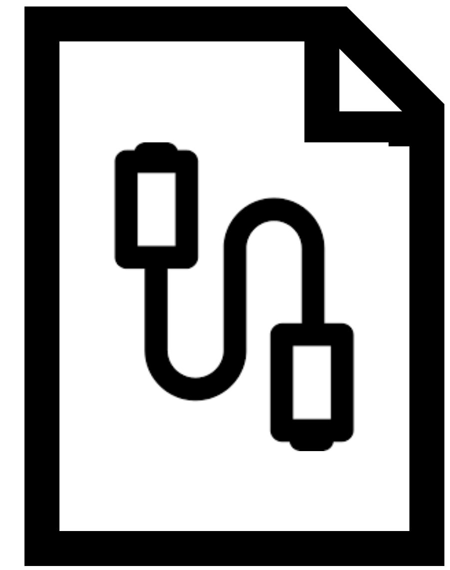


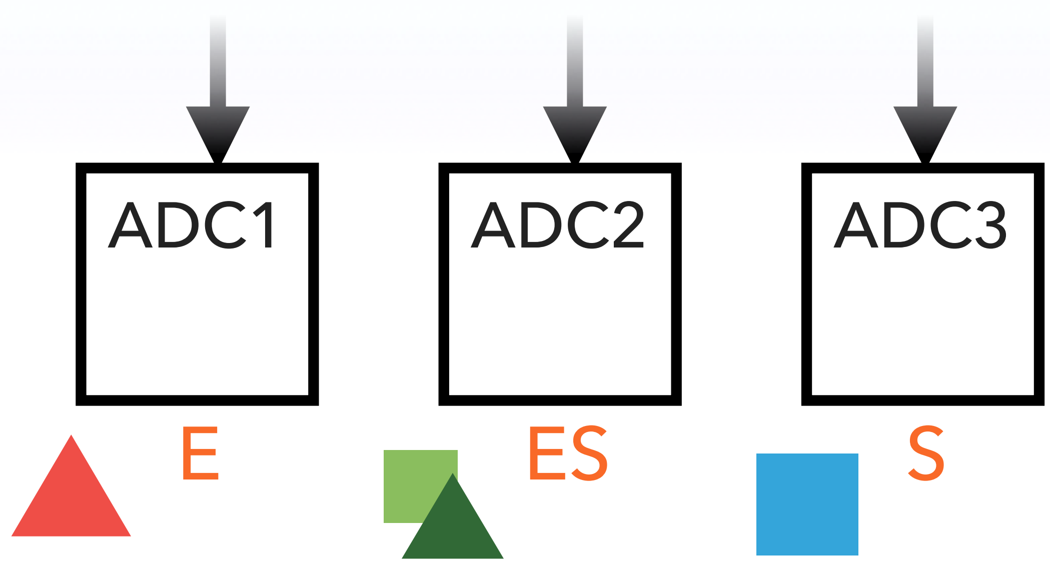


ANALOG DEVICE CONFIGURATION REVISITED

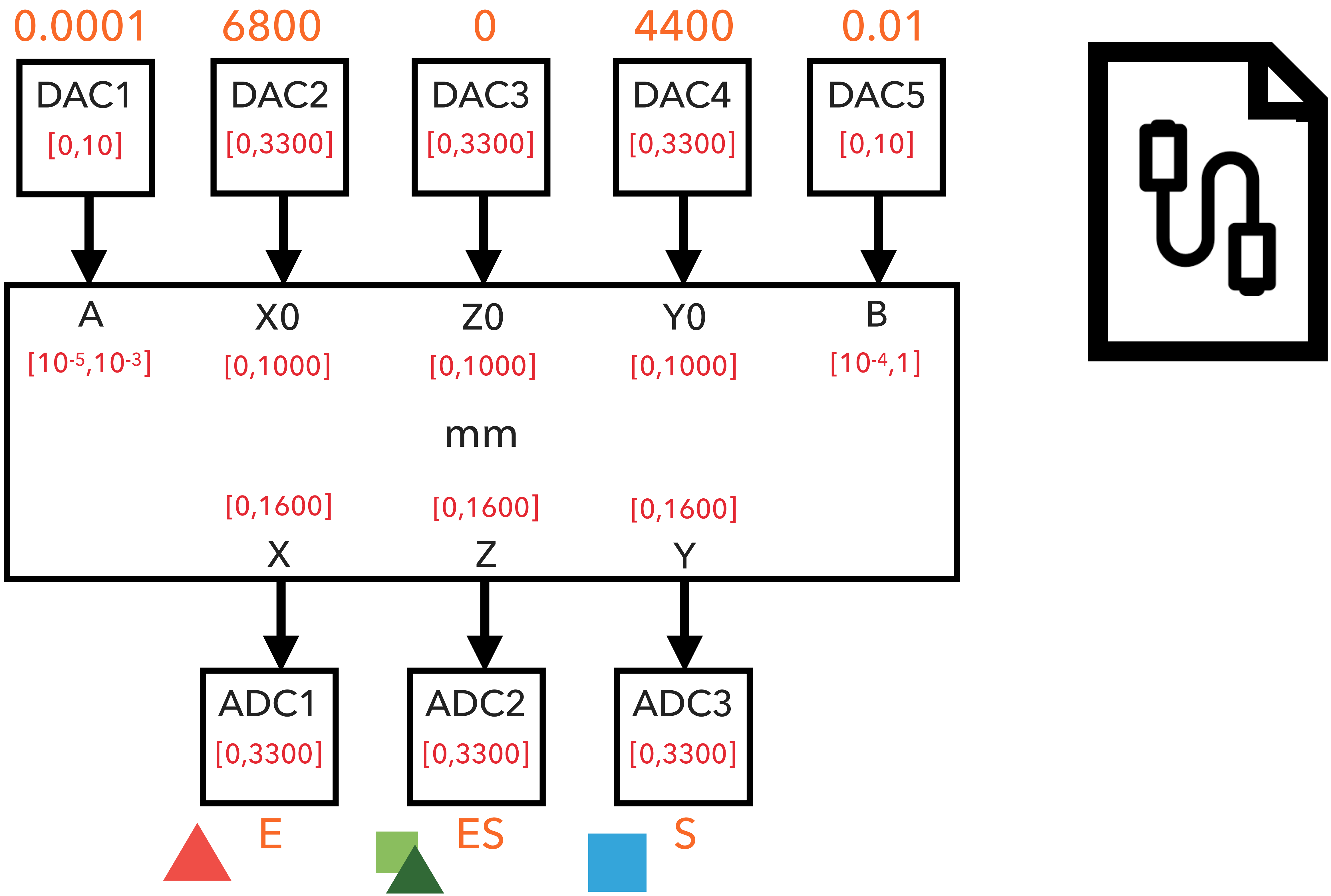
Expected Simulation Dynamics



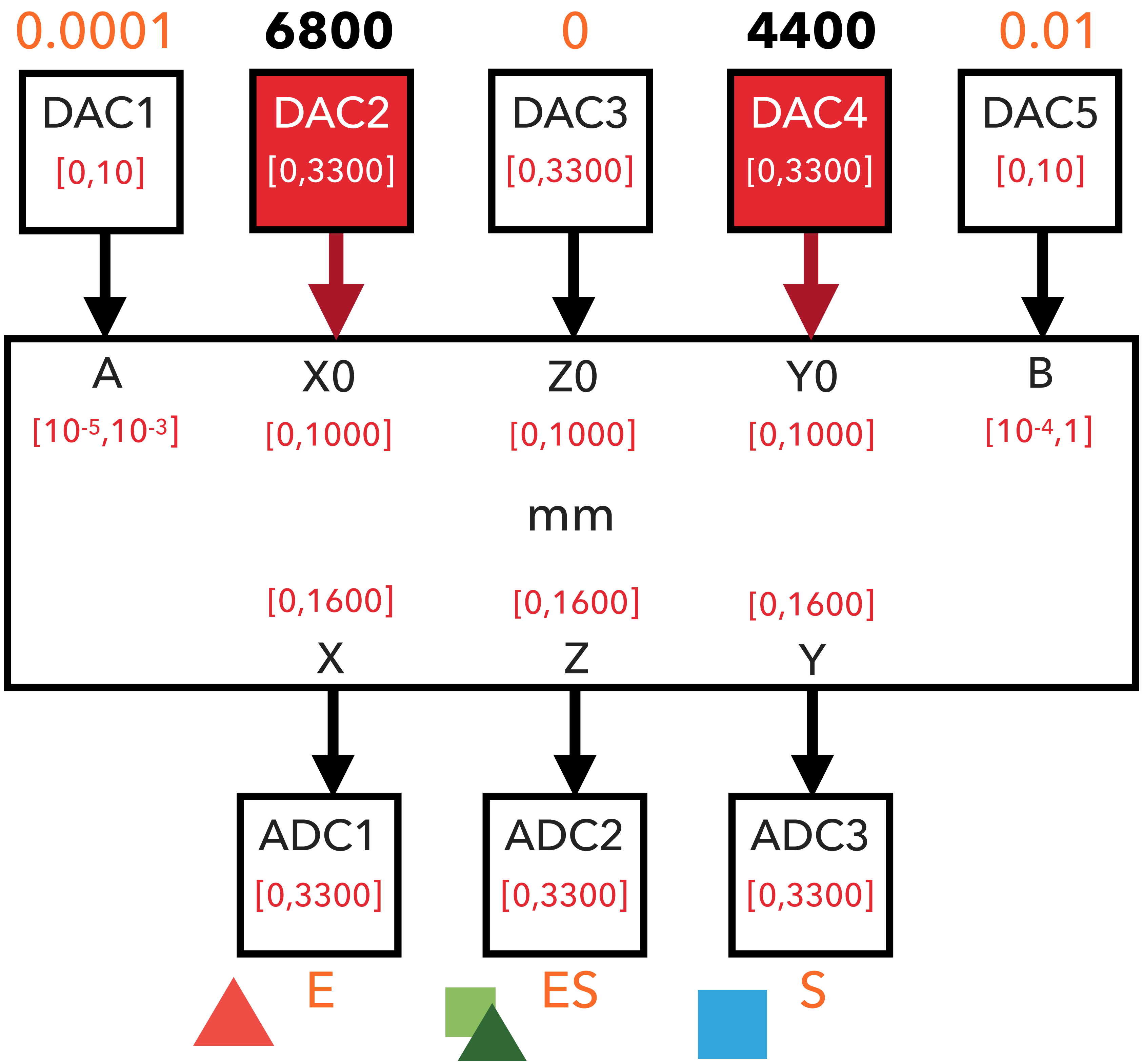




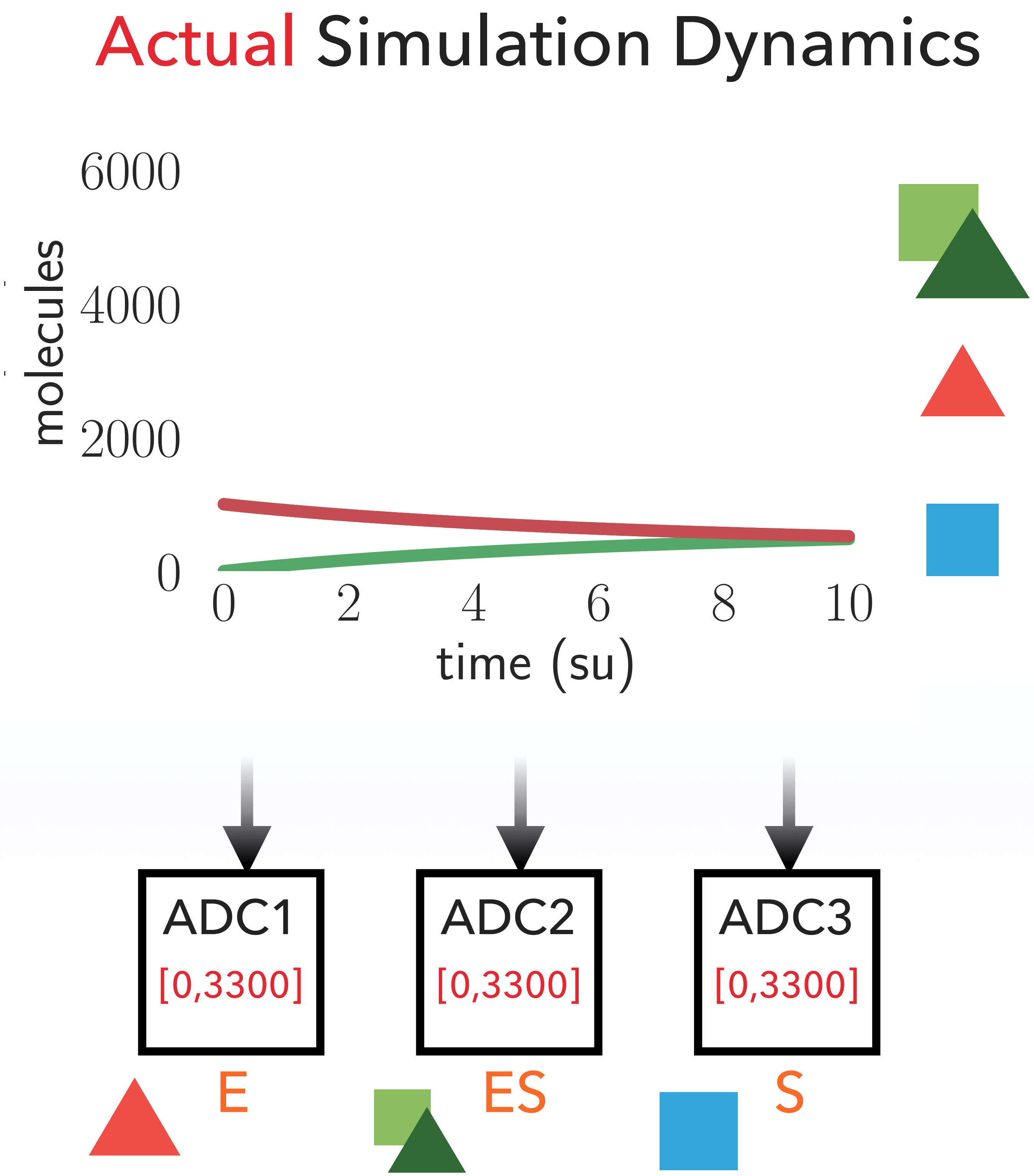
ANALOG DEVICE CONFIGURATION REVISITED

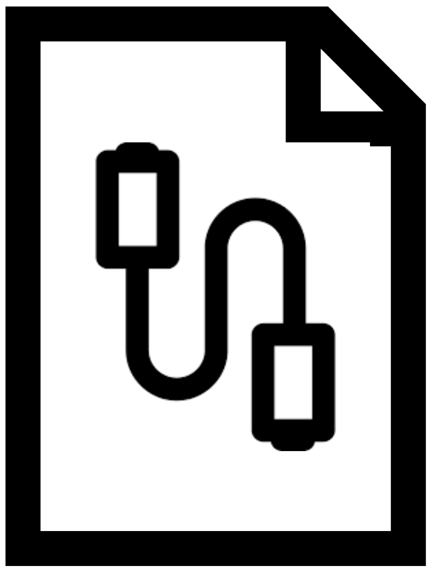


ANALOG DEVICE CONFIGURATION REVISITED

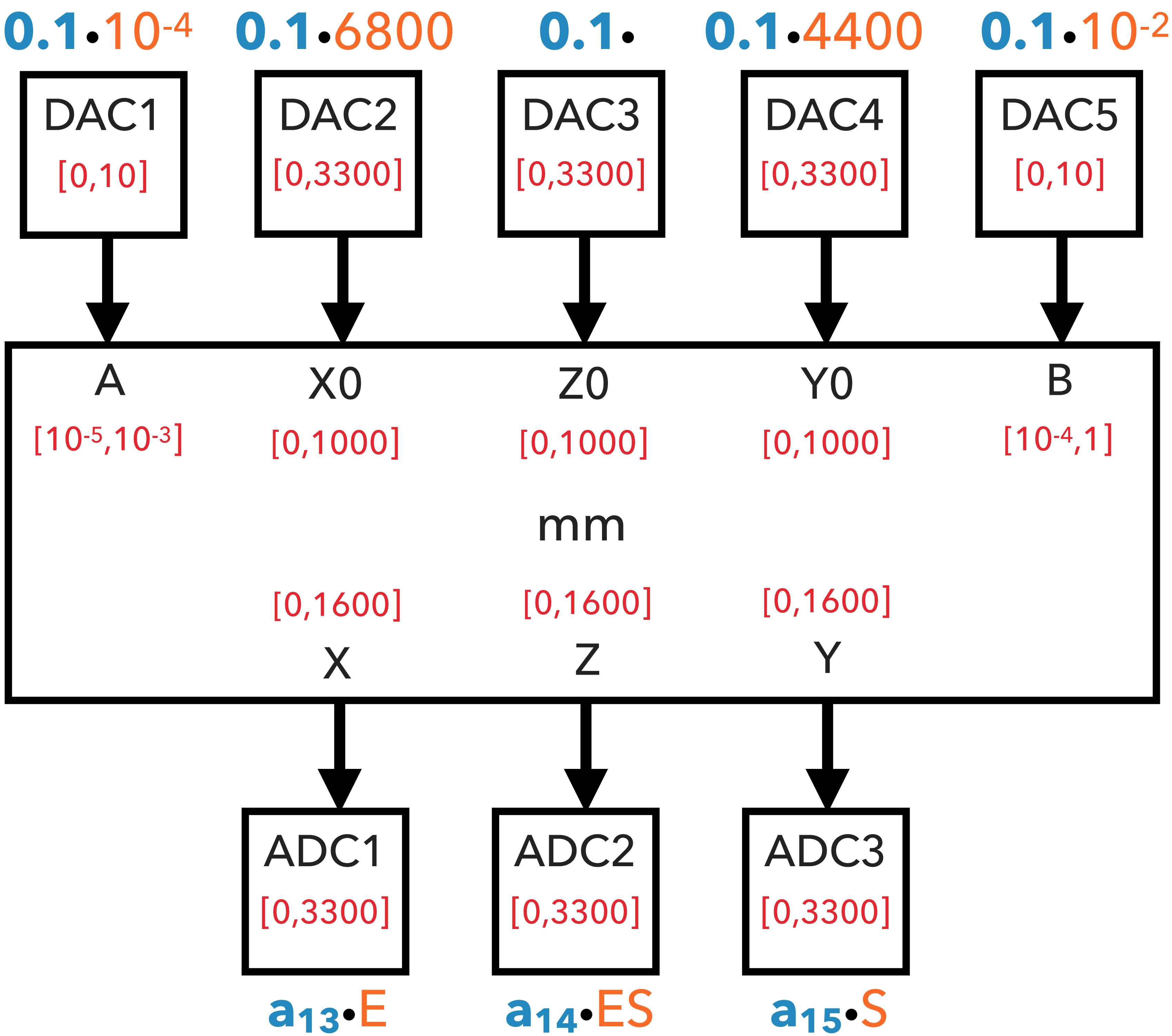


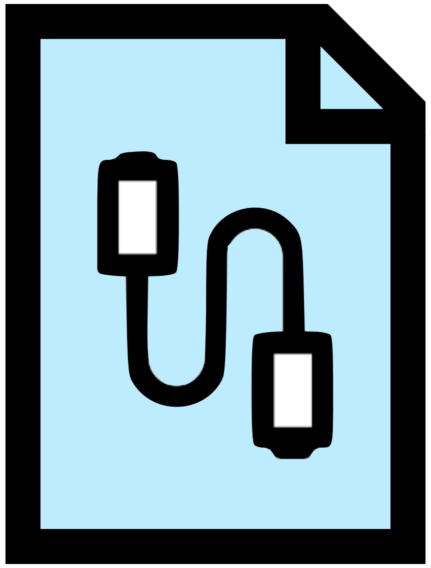
ANALOG DEVICE CONFIGURATION REVISITED



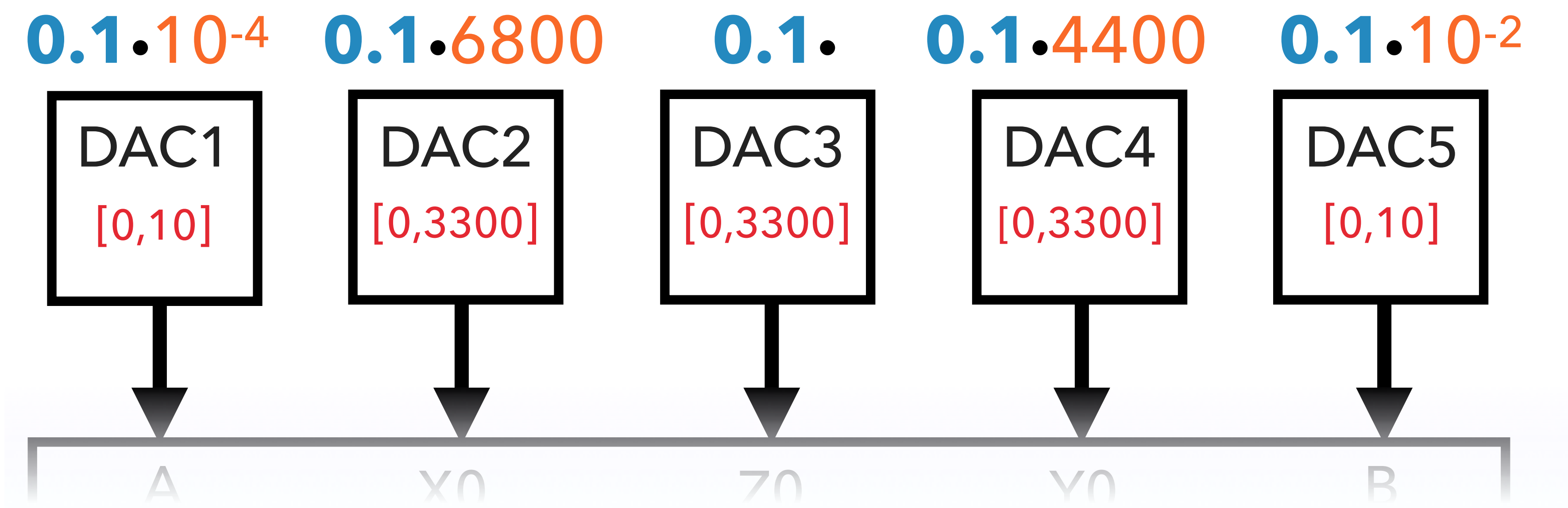


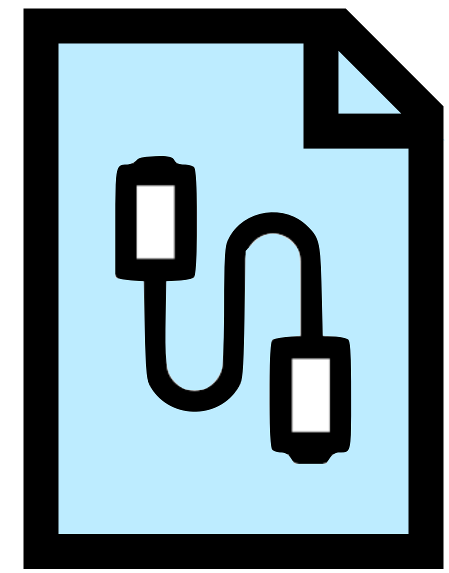
UNIFORMLY SCALED ANALOG DEVICE CONFIGURATION





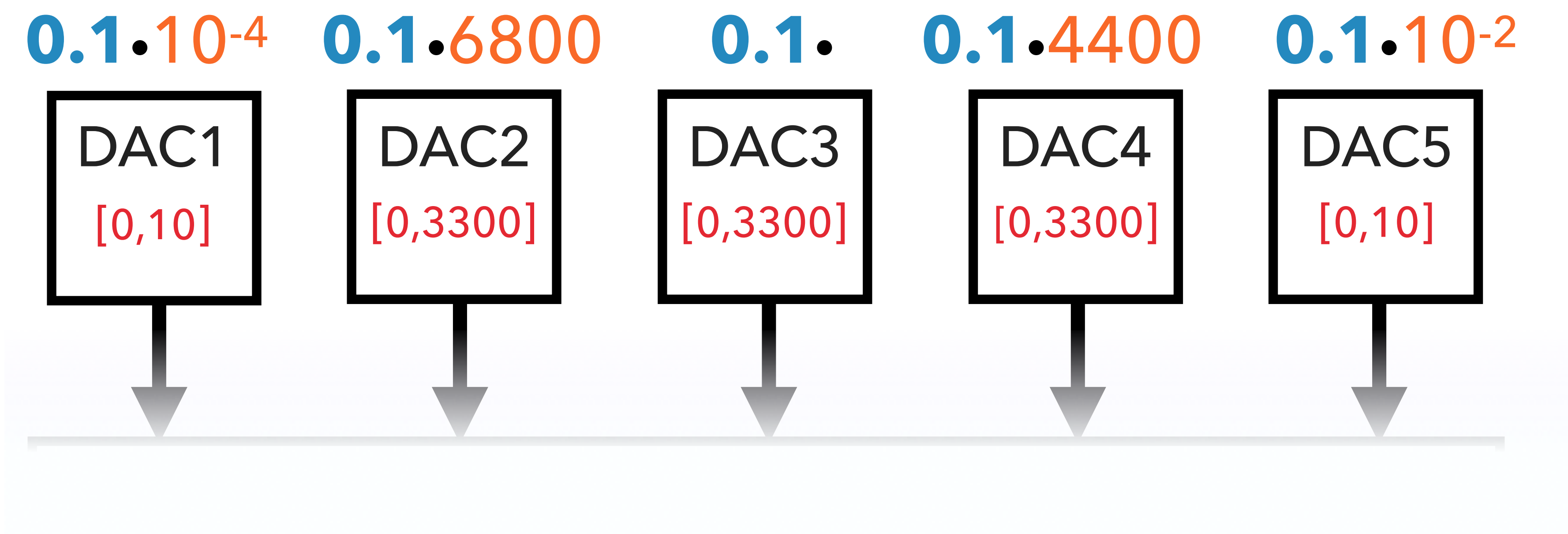
UNIFORMLY SCALED ANALOG DEVICE CONFIGURATION

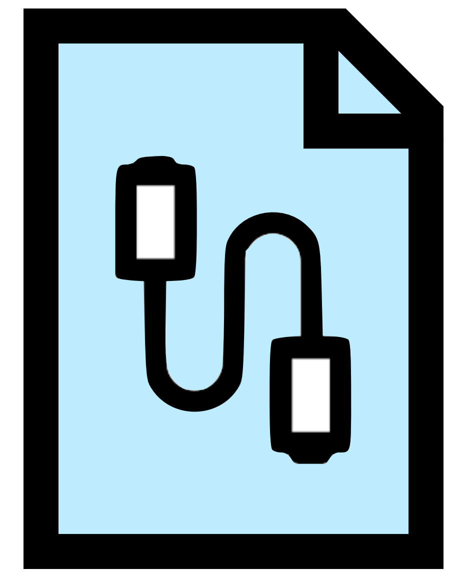




DOES NOT WORK!

UNIFORMLY SCALED ANALOG DEVICE CONFIGURATION





DOES NOT WORK!

SCALED SIGNAL CHANGES SIMULATION

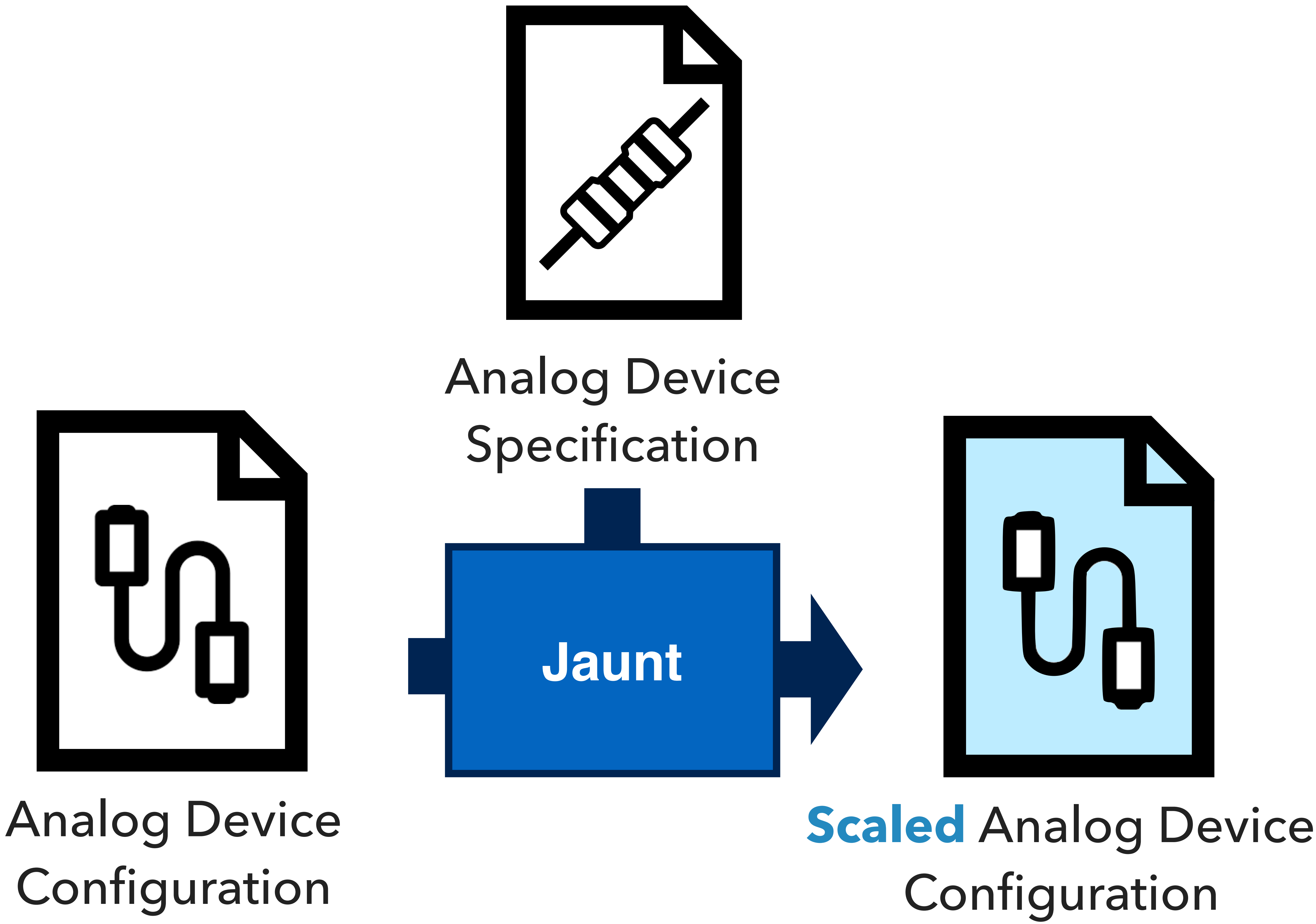
ORIGINAL SIMULATION NOT RECOVERABLE

TALK OUTLINE

- ~~Background: overview of compilation problem~~
- ~~Arco Compiler[1]: automatically configure analog devices to simulate dynamical systems.~~
- Jaunt Solver[2]: automatically scales dynamical systems to execute analog hardware with operating range constraints.
- Closing Remarks

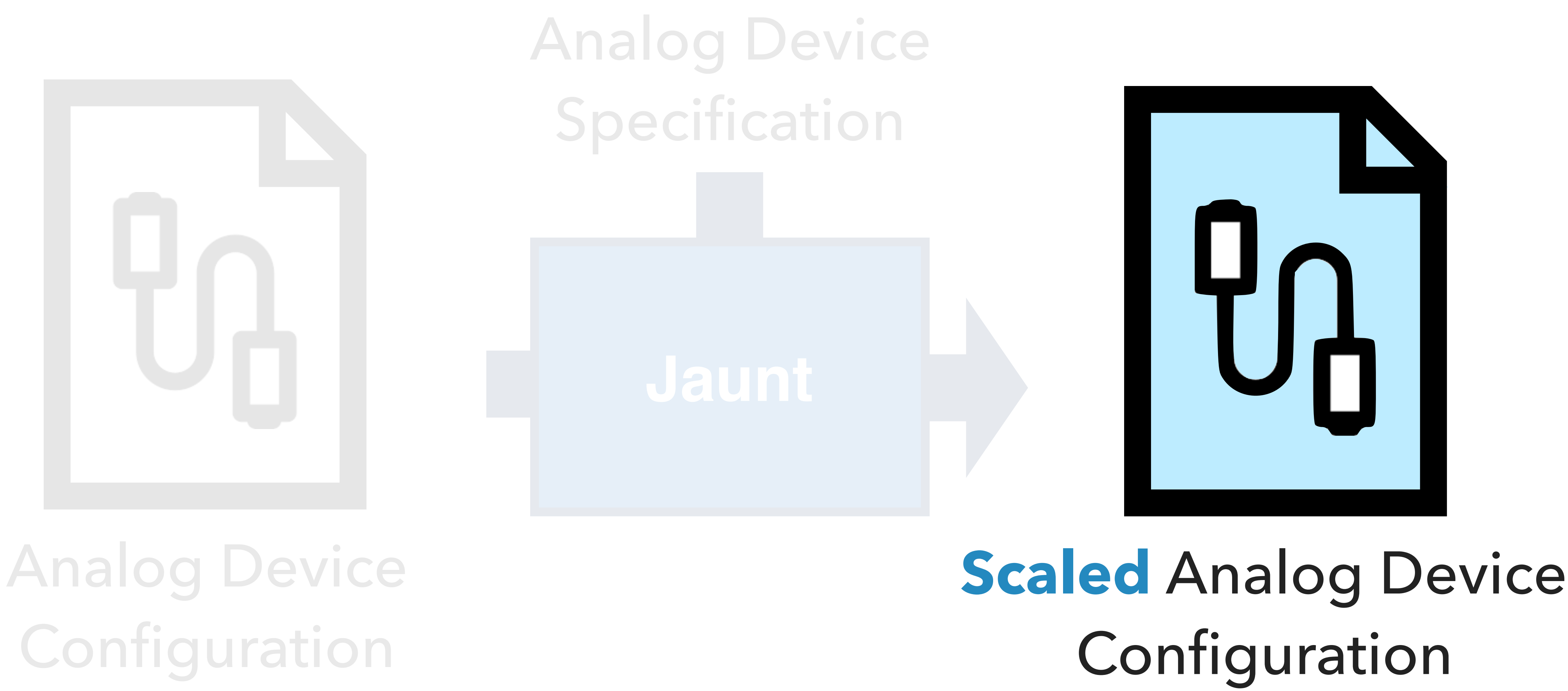
1. Configuration Synthesis for Programmable Analog Devices with Arco. Sara Achour, Rahul Sarpeshkar and Martin Rinard. June 2016. PLDI 2016.
2. Time Dilation and Contraction for Programmable Analog Devices with Jaunt. Sara Achour and Martin Rinard. December 2017. ASPLOS 2018.

JAUNT SOLVER

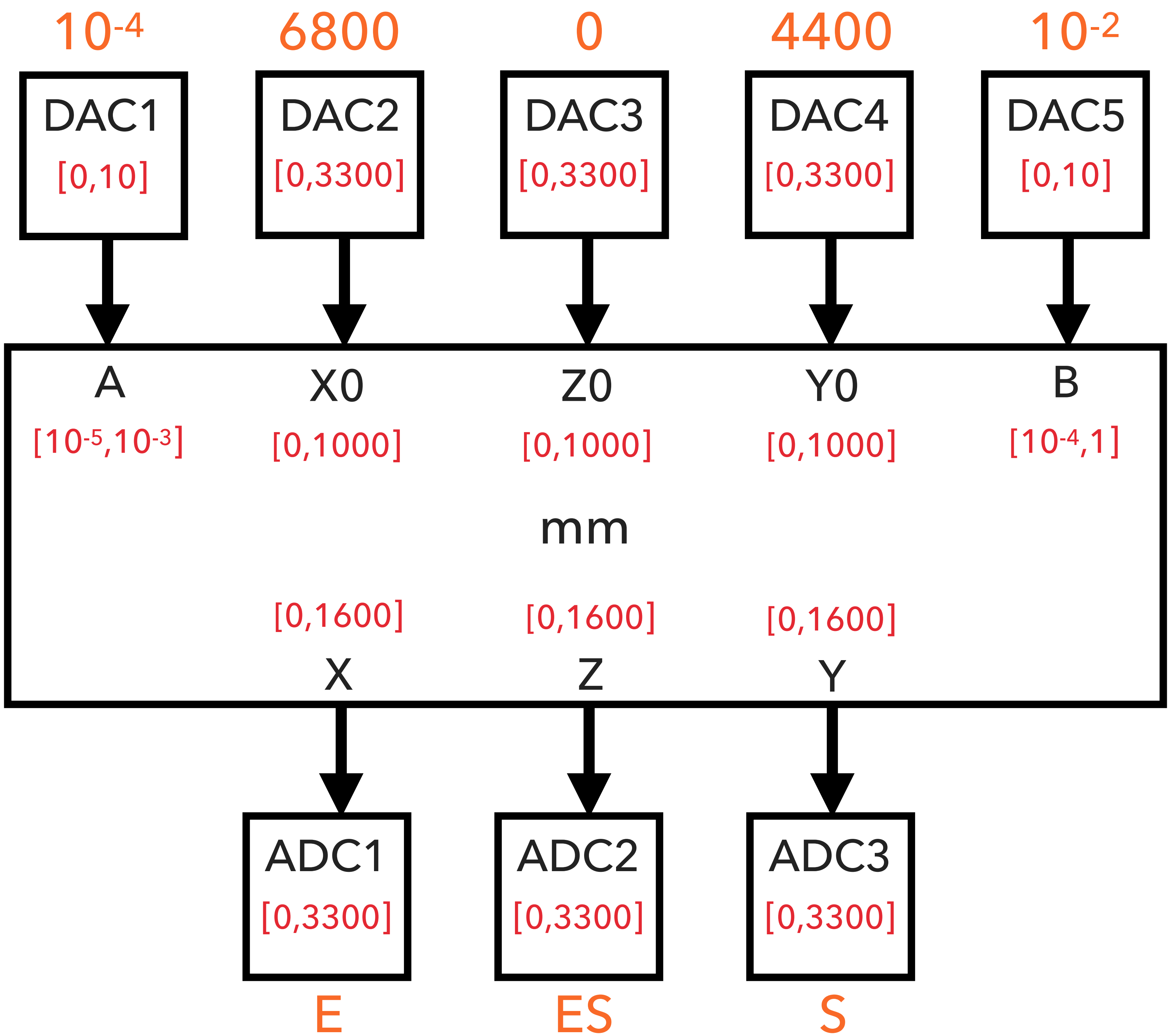

Analog Device Specification
Analog Device Configuration
Jaunt
Scaled Analog Device Configuration

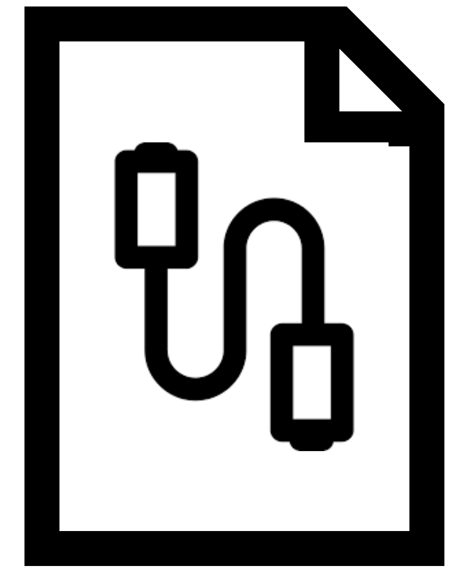
physically realizable: signals within port operating ranges

recoverable: recover original simulation at ADCs

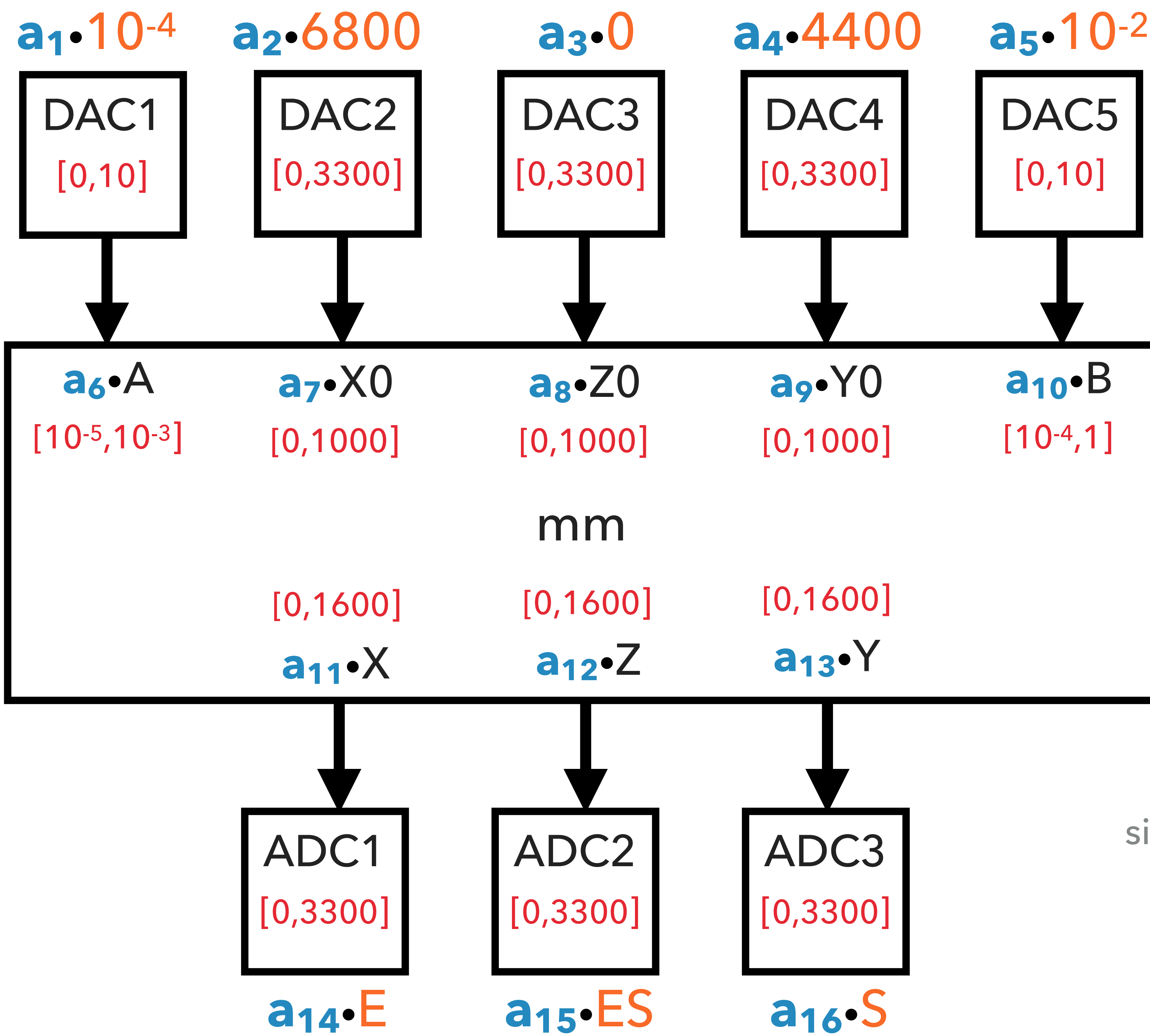


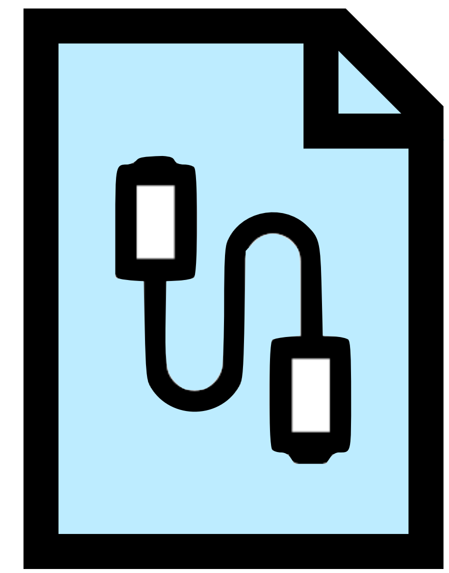
ANALOG DEVICE CONFIGURATION





SCALED ANALOG DEVICE CONFIGURATION

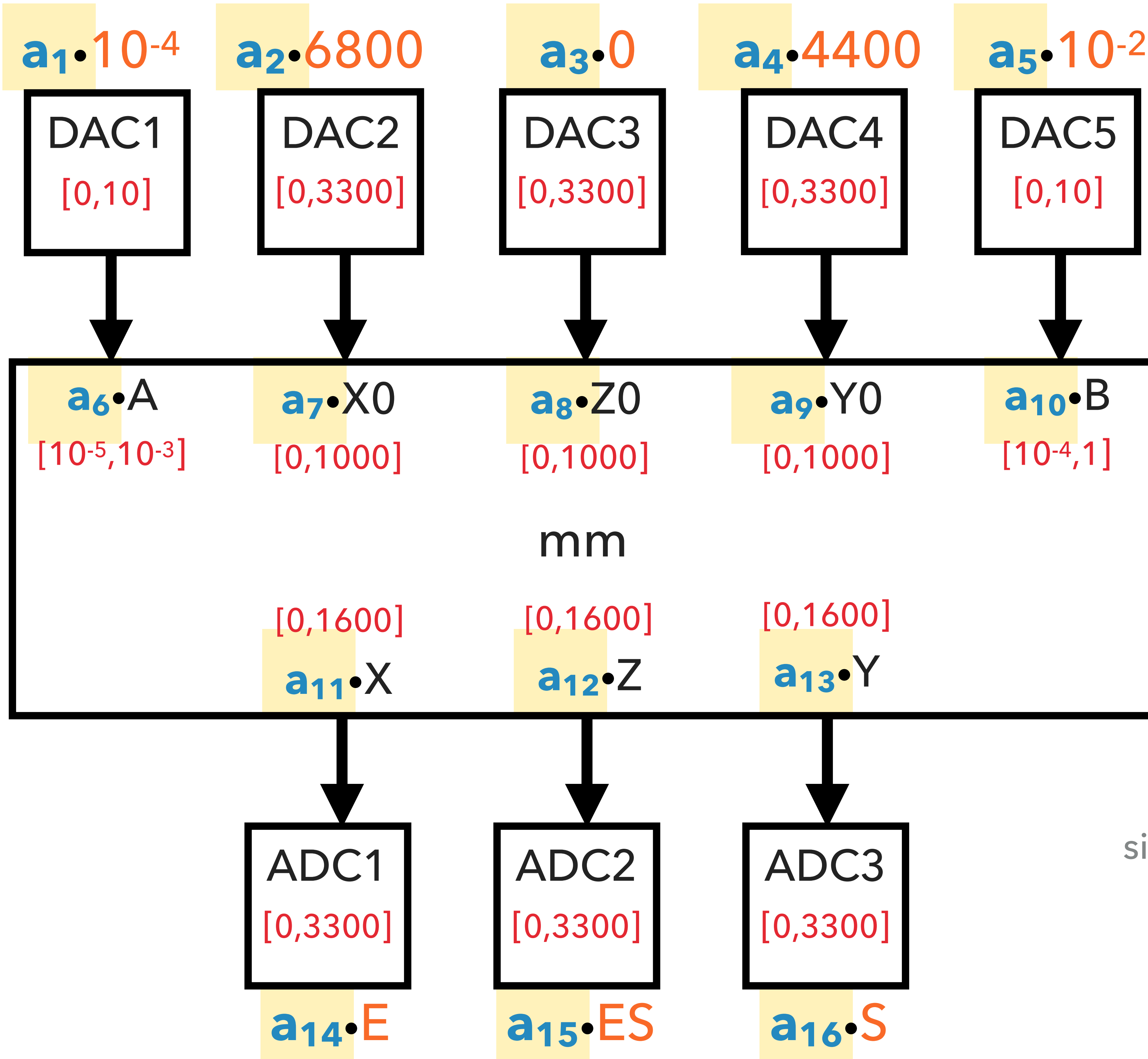


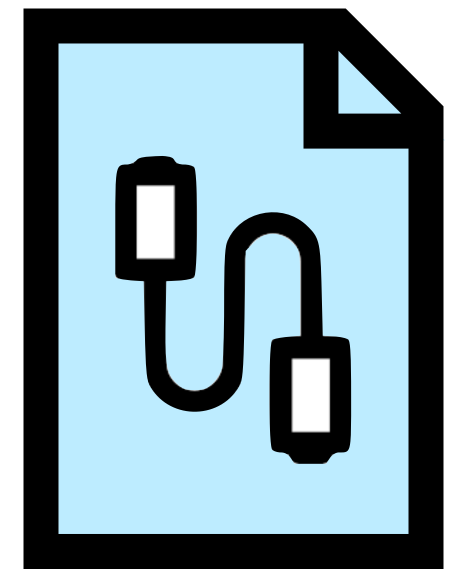


simulation speed

T

SCALED ANALOG DEVICE CONFIGURATION

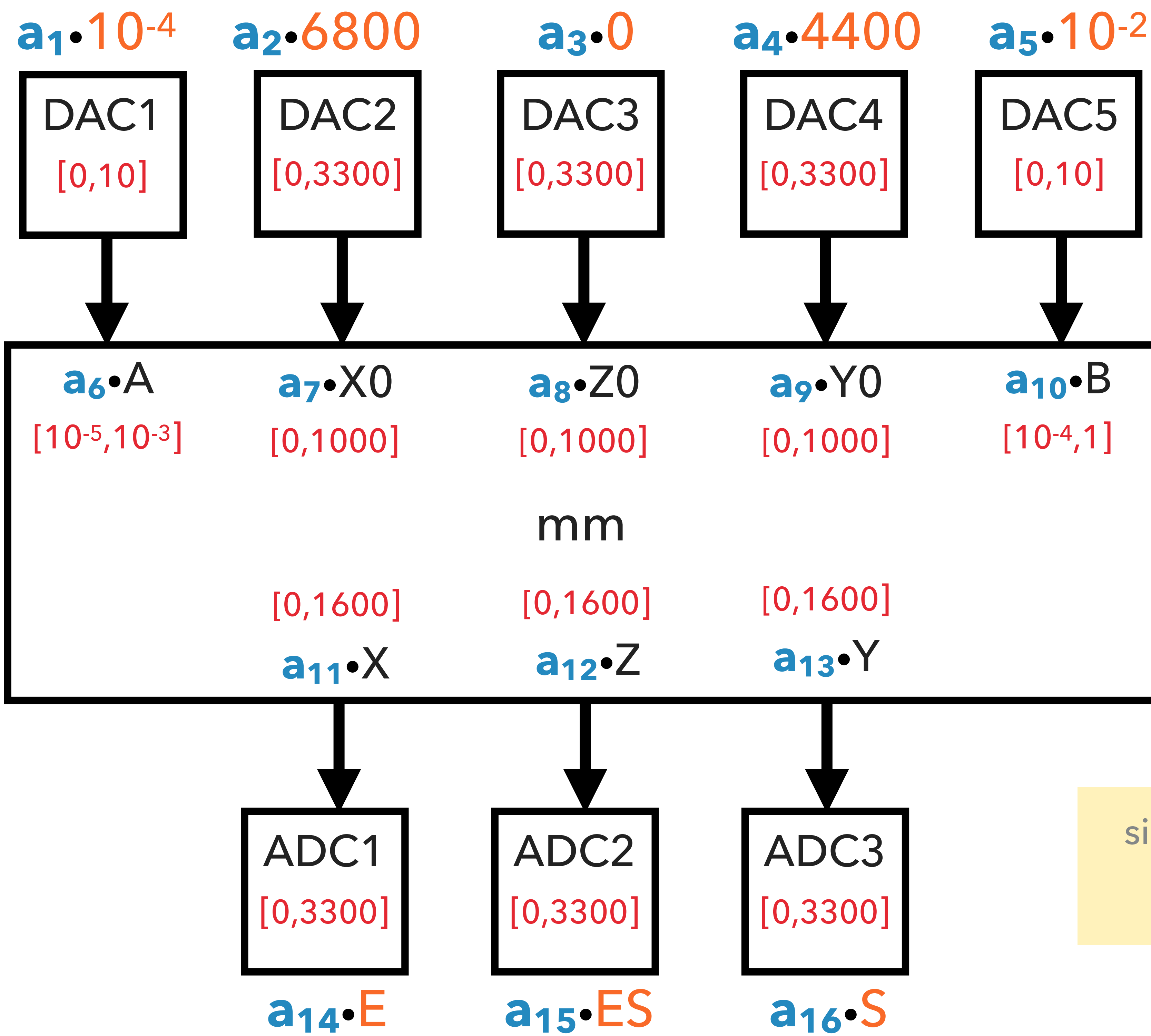


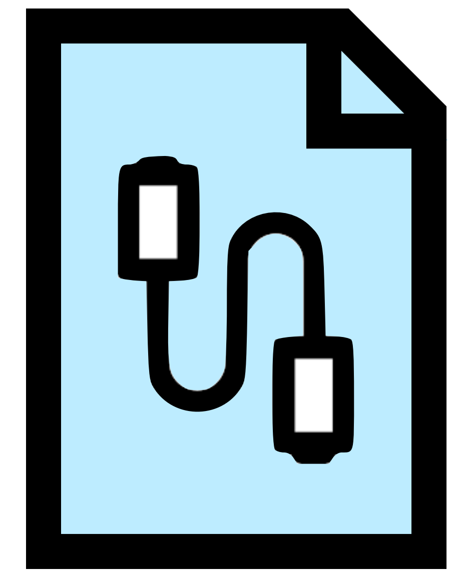


simulation speed

T

SCALED ANALOG DEVICE CONFIGURATION

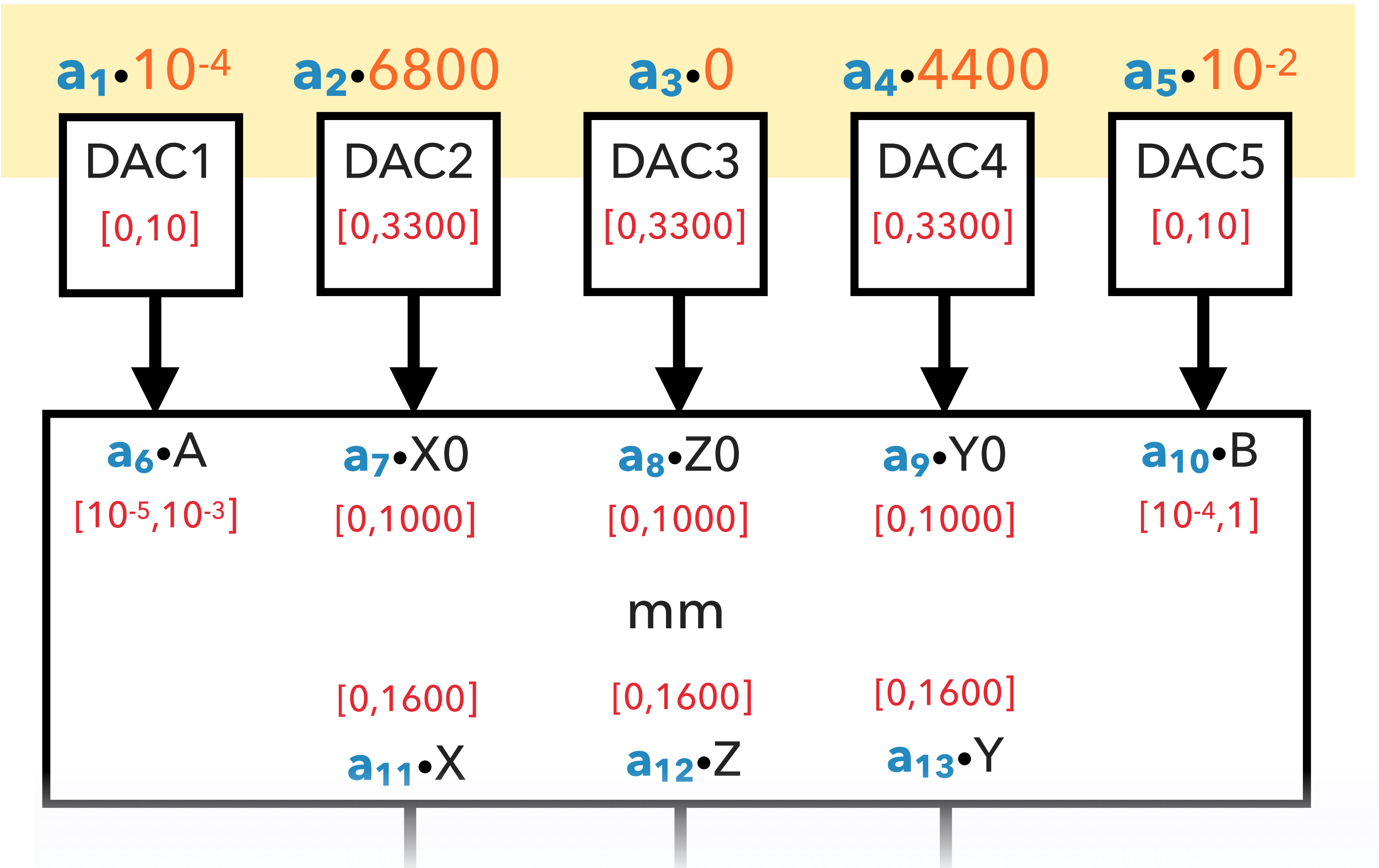


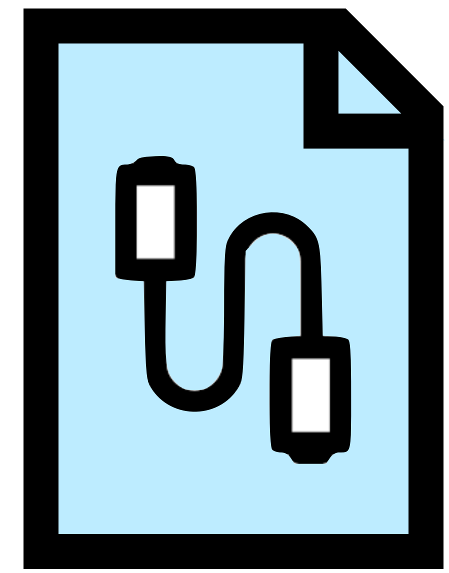


simulation speed

T

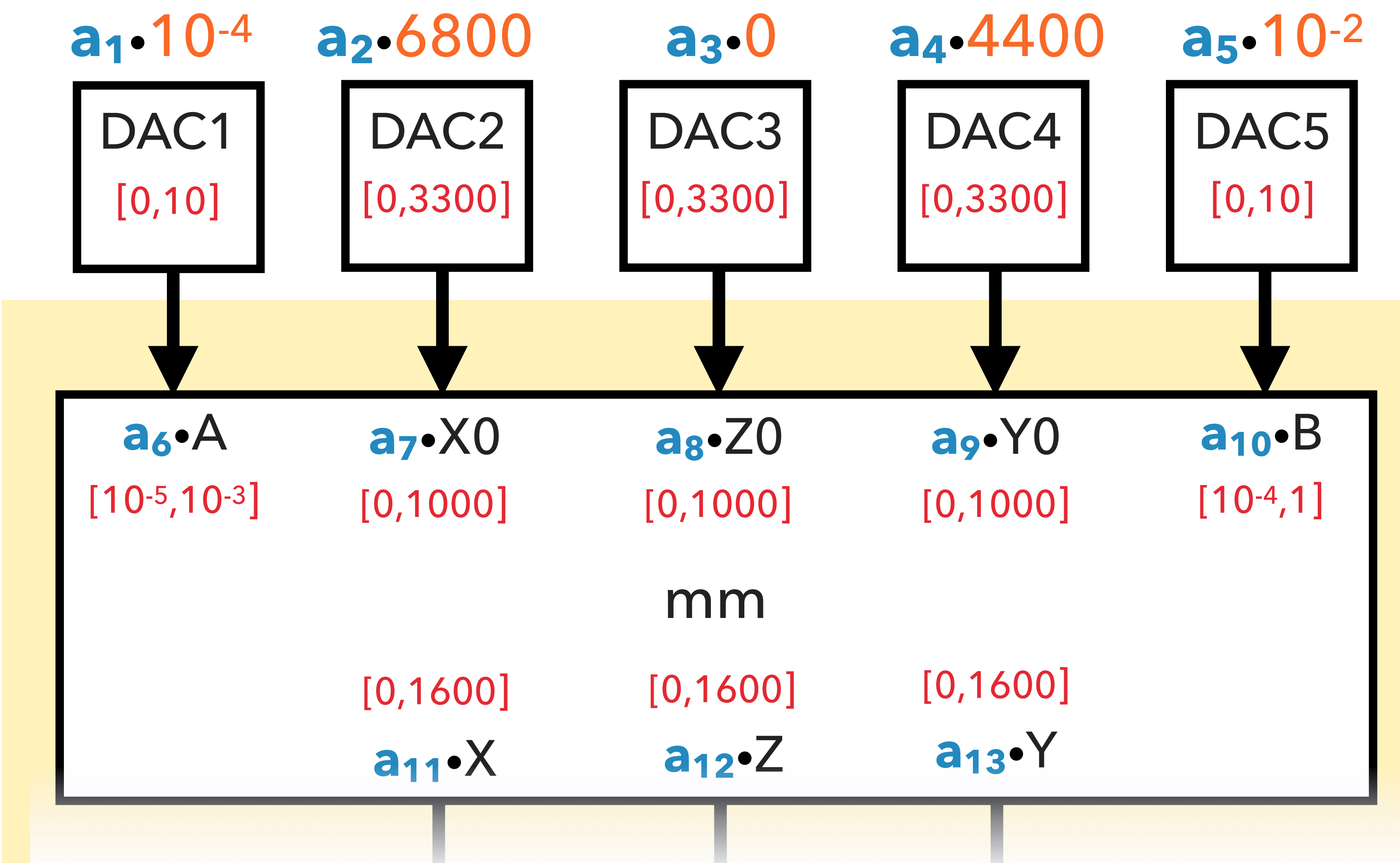
SCALED ANALOG DEVICE CONFIGURATION

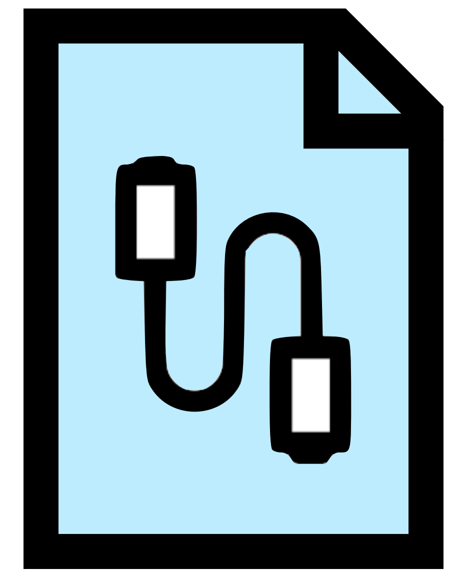




multiply parameters by scaling factors

SCALED ANALOG DEVICE CONFIGURATION

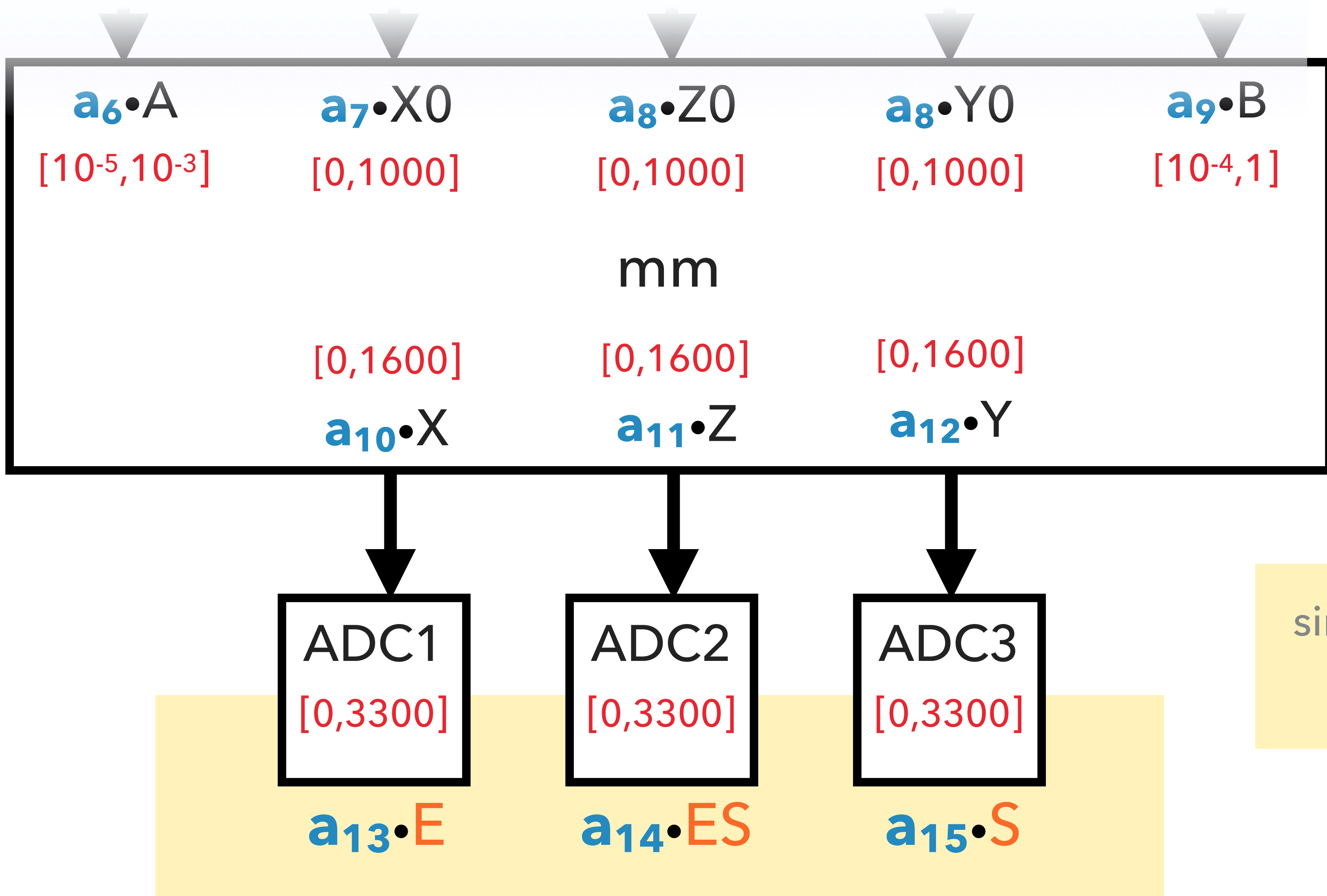


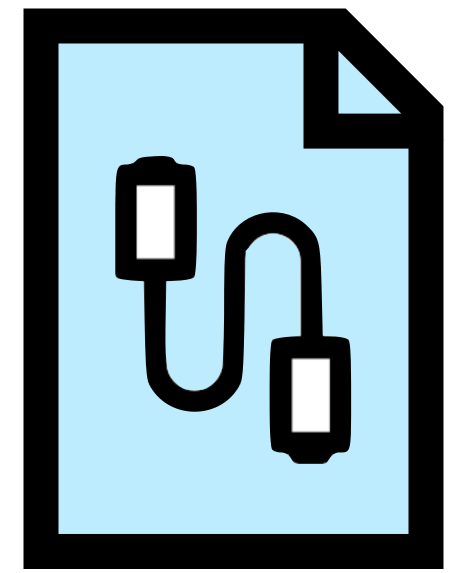


physically realizable: signals in port operating ranges

SCALED ANALOG DEVICE CONFIGURATION

- Recover Simulation:
  - divide ADC samples by a_{13}-a_{15}
  - multiply hardware time by T





simulation speed

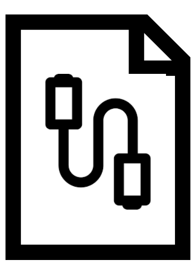
T

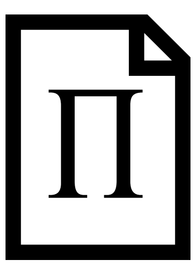
JAUNT SOLVER

- **Objective**: Find numerical assignments for scaling factors that produces fastest simulation
  - Scaling Factors: $\tau, a_1, \ldots, a_{15}$
  - Values: real numbers > 0

JAUNT SOLVER

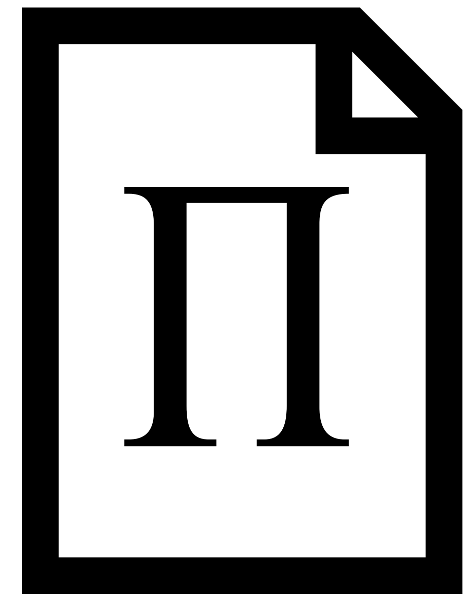
- **Objective**: Find numerical assignments for scaling factors that produces fastest simulation
  - Scaling Factors: $\tau, a_1, \ldots, a_{15}$
  - Values: real numbers > 0
- **Geometric Program**: Convex optimization problem
  - Device Configuration → Geometric Program





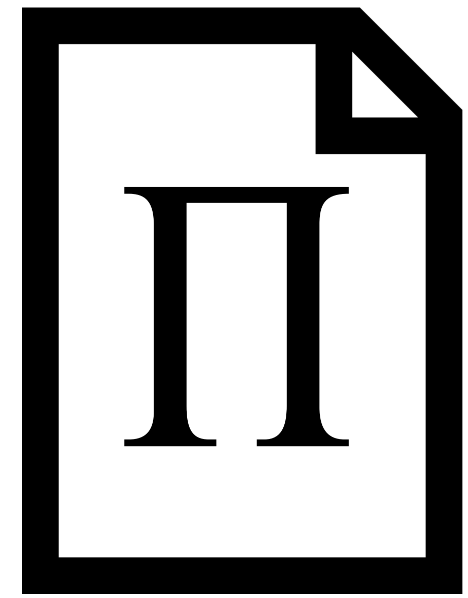
GEOMETRIC PROGRAM GENERATION

- Maximize τ subject to:
  - Factor Constraints
  - Sampling Constraints
  - Connection Constraints
  - Operating Range Constraints



GEOMETRIC PROGRAM GENERATION

- Maximize τ subject to:
    - Factor Constraints
    - Sampling Constraints
    - Connection Constraints
    - Operating Range Constraints



given block with scaled input signals:

→ original output signal is recoverable from scaled output signal

FACTOR CONSTRAINTS

simulation speed

T

a_6•A	a_7•X0	a_8•Z0	a_9•Y0	a_{10}•B
$[10^{-5},10^{-3}]$	[0,1000]	[0,1000]	[0,1000]	$[10^{-4},1]$
		mm		
	[0,1600]	[0,1600]	[0,1600]	
	a_{11}•X	a_{12}•Z	a_{13}•Y	

$$a_{12} \cdot Z = \int [a_6 a_{11} a_{13} \cdot A \cdot X \cdot Y - a_9 a_{12} \cdot B \cdot Z] \tau^{-1} \cdot dt$$

FACTOR CONSTRAINTS

simulation speed

T

a_6•A	a_7•X0	a_8•Z0	a_9•Y0	a_{10}•B
$[10^{-5},10^{-3}]$	[0,1000]	[0,1000]	[0,1000]	$[10^{-4},1]$
		mm		
	[0,1600]	[0,1600]	[0,1600]	
	a_{11}•X	a_{12}•Z	a_{13}•Y	

$$a_{12} \cdot Z = \int a_6 a_{11} a_{13} \cdot [A \cdot X \cdot Y - \cdot B \cdot Z] \tau^{-1} \cdot dt$$

$$a_6 a_{11} a_{13} = a_9 a_{12}$$

FACTOR CONSTRAINTS

simulation speed

T

a_6•A	a_7•X0	a_8•Z0	a_9•Y0	a_{10}•B
[10^{-5},10^{-3}]	[0,1000]	[0,1000]	[0,1000]	[10^{-4},1]
		mm		
	[0,1600]	[0,1600]	[0,1600]	
	a_{11}•X	a_{12}•Z	a_{13}•Y	

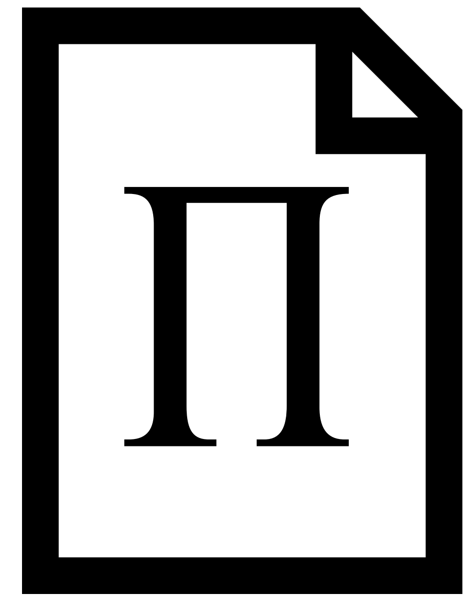
$$a_6 a_{11} a_{13} \tau^{-1}[Z = \int \cdot [A \cdot X \cdot Y - \cdot B \cdot Z] \cdot dt]$$

$$a_6 a_{11} a_{13} = a_9 a_{12}$$

$$a_6 a_{11} a_{13} \tau^{-1} = a_{12}$$

GEOMETRIC PROGRAM GENERATION

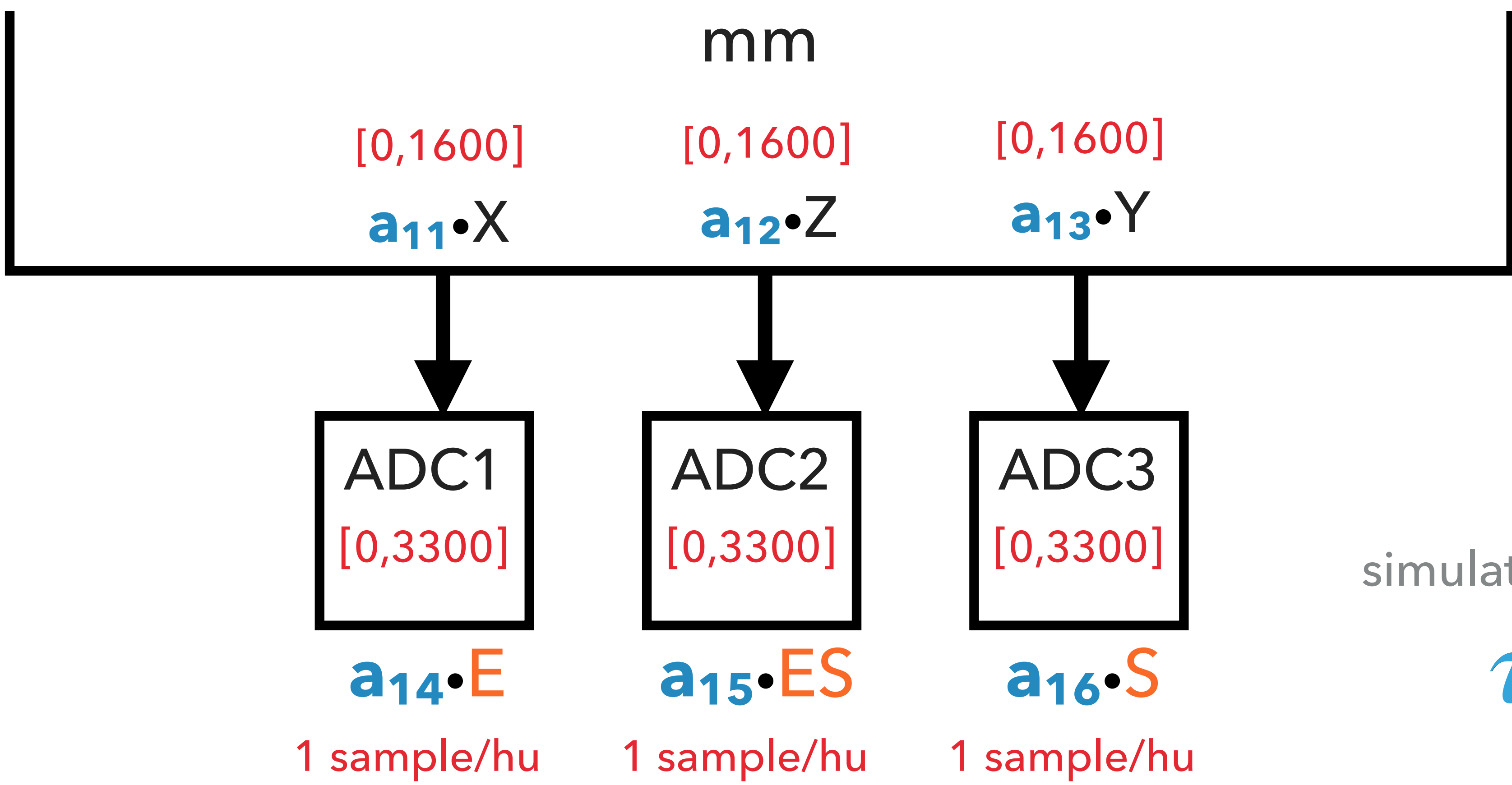
- Maximize τ subject to:
  - Factor Constraints
  - Sampling Constraints
  - Connection Constraints
  - Operating Range Constraints

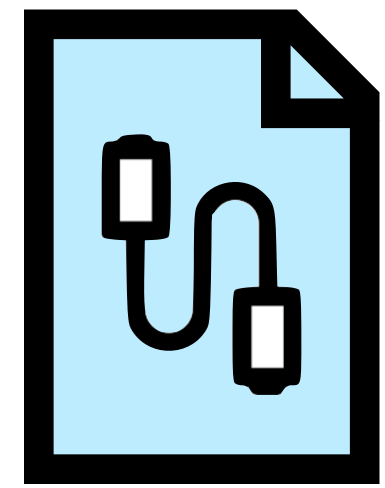


given DAC/ADC:

→ ensure simulation is executed slowly enough for adequate sampling

SAMPLE CONSTRAINTS





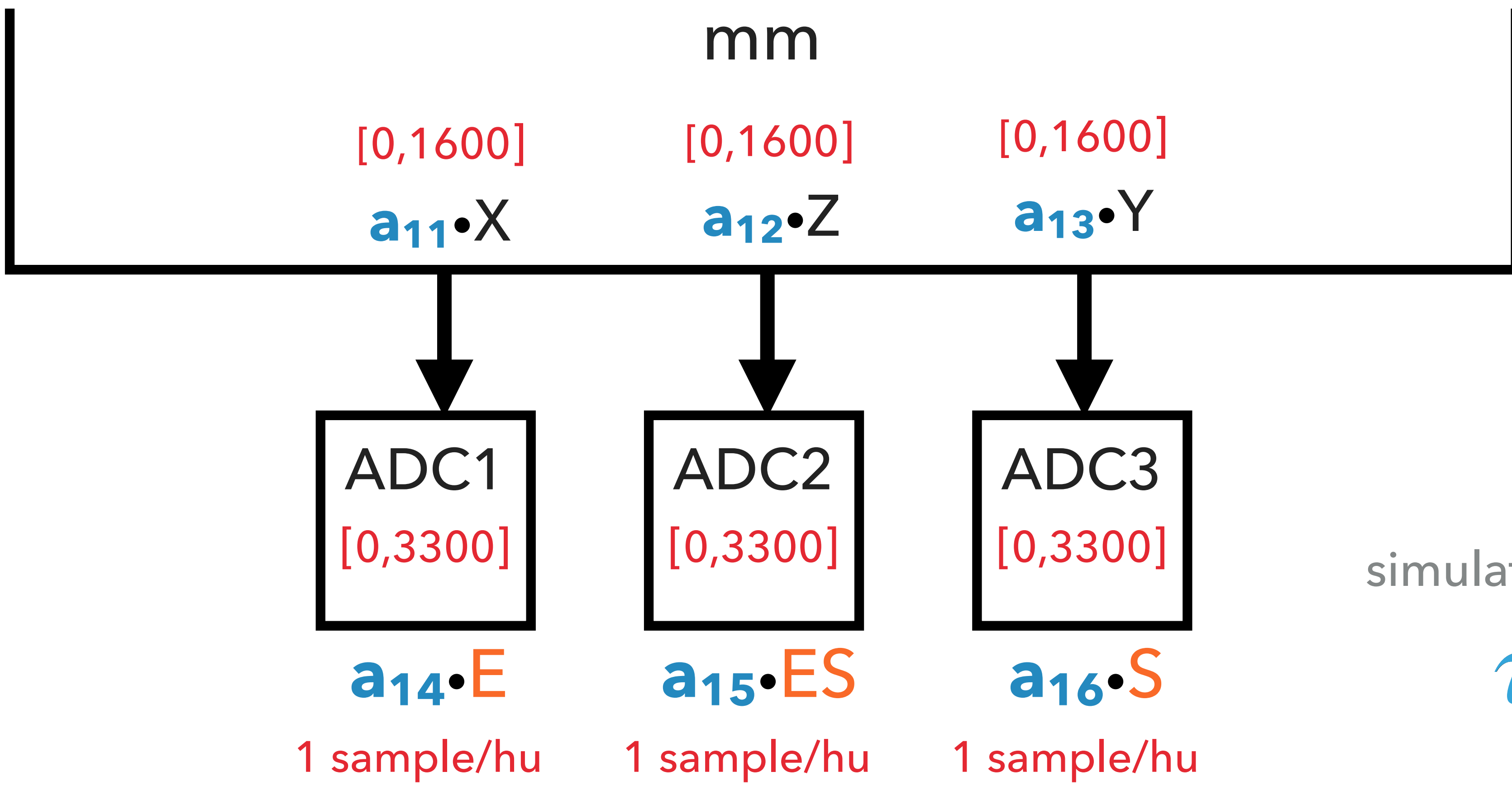
simulation speed

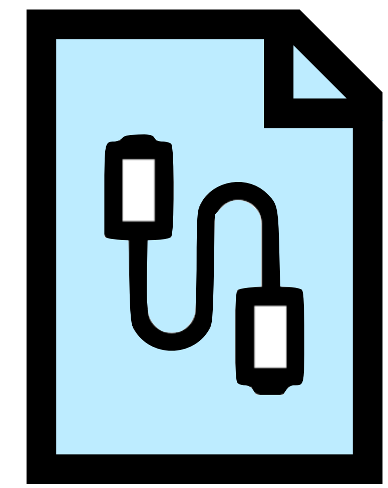
τ : hu $\rightarrow$ su

$$\boxed{2\ \text{sample/su}} \leq 1\ \text{sample/hu} \cdot \tau^{-1}$$

minimum number of samples / simulation time unit

SAMPLE CONSTRAINTS





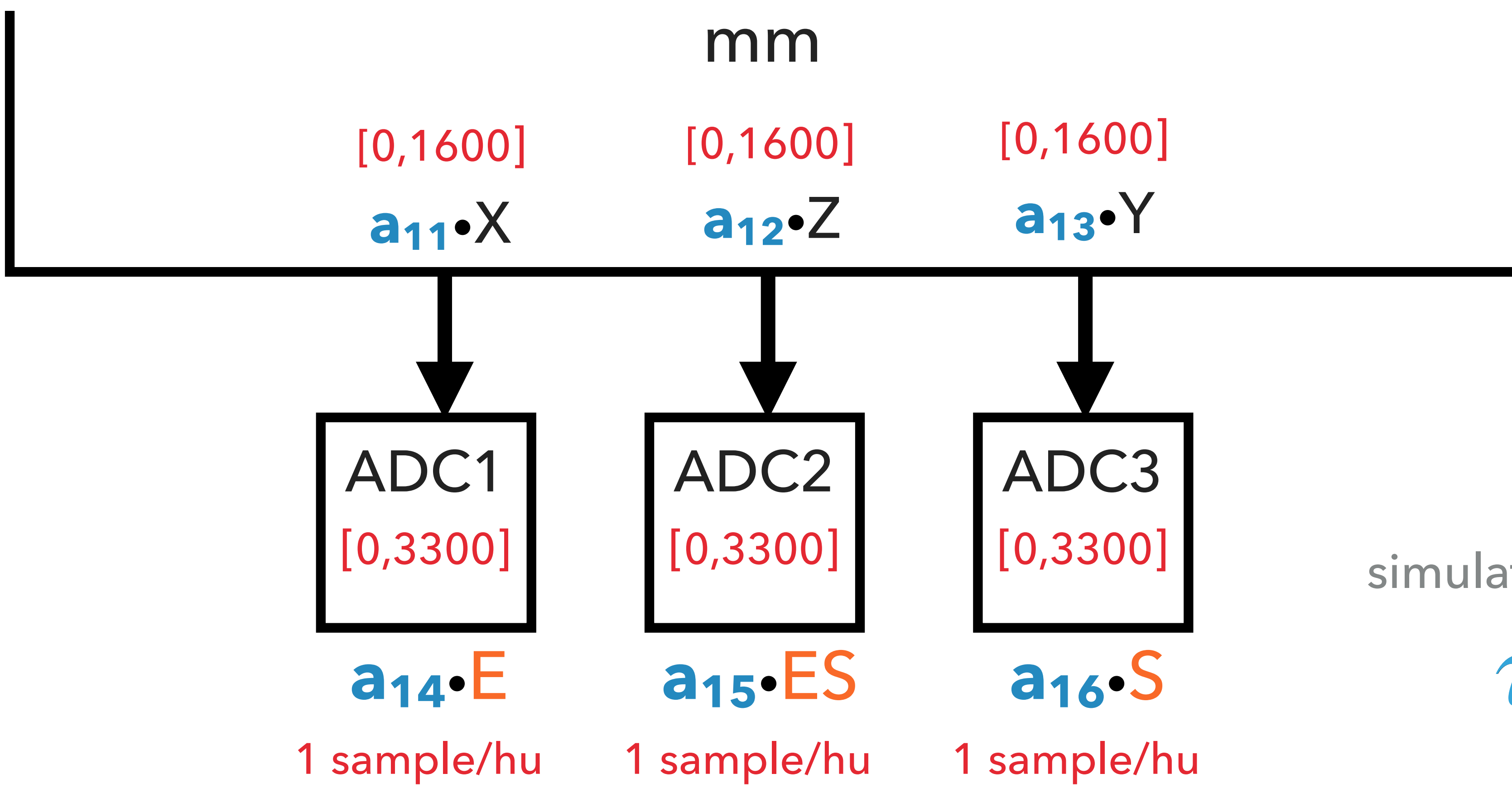
simulation speed

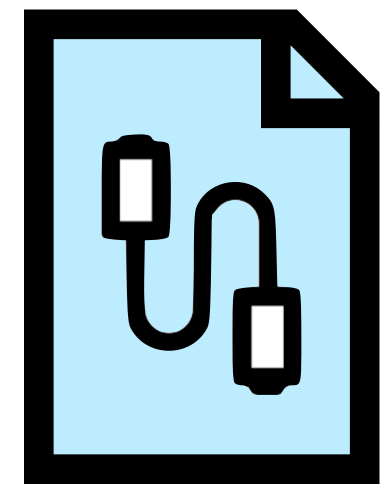
$\tau : \mathrm{hu} \to \mathrm{su}$

$$2 \ \text{sample/su} \leq 1 \ \text{sample/hu} \cdot \tau^{-1}$$

number of samples in one hardware time unit

SAMPLE CONSTRAINTS





simulation speed

$$\tau : \text{hu} \rightarrow \text{su}$$

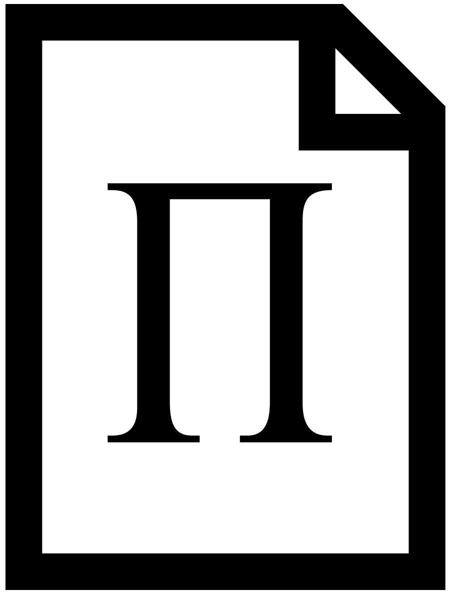
$$\tau^{-1} : \frac{\text{hu}}{\text{su}}$$

$$2 \ \text{sample/su} \leq 1 \ \text{sample/hu} \cdot \tau^{-1}$$

number of samples in one hardware time unit

GEOMETRIC PROGRAM GENERATION

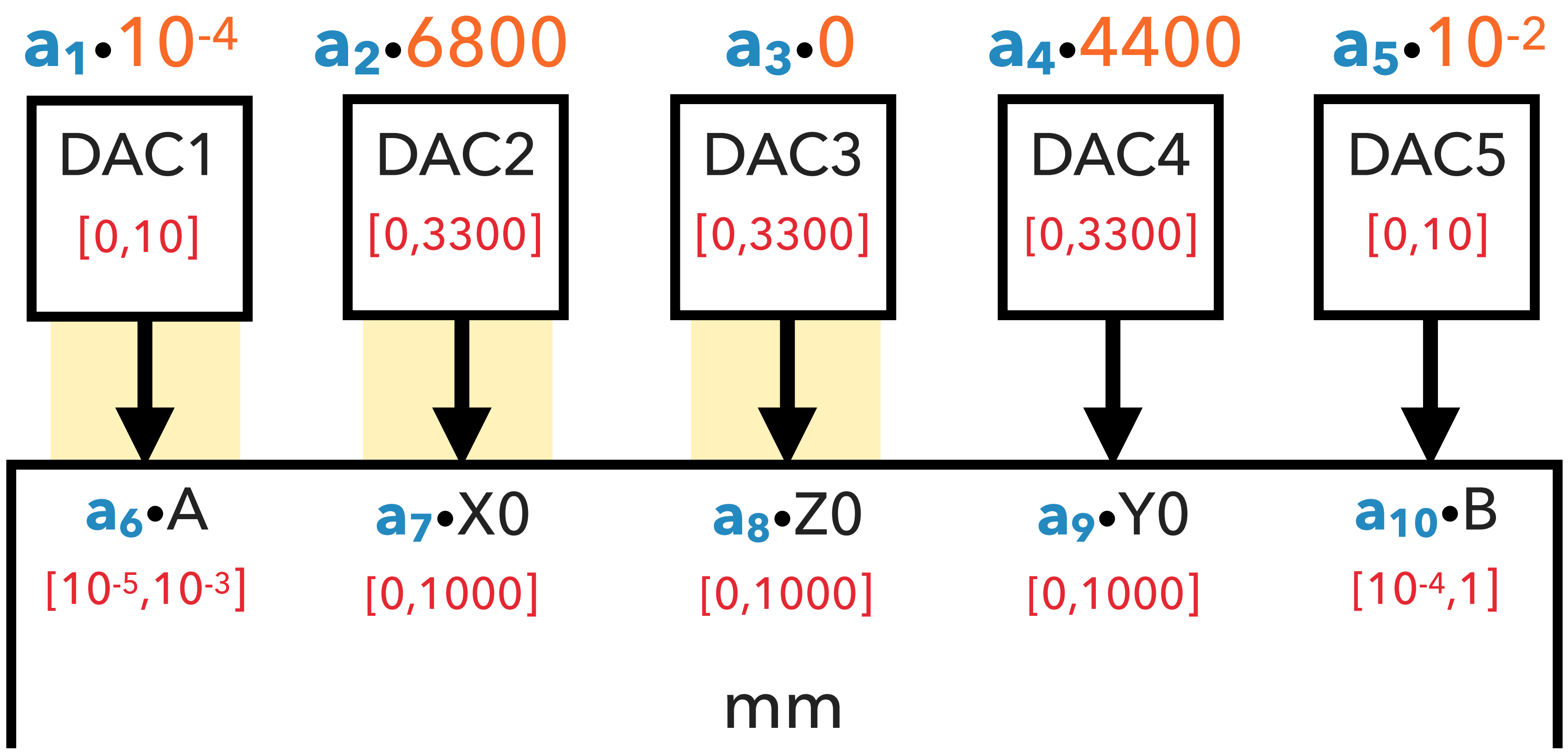
- Maximize τ subject to:
  - Factor Constraints
  - Sampling Constraints
  - Connection Constraints
  - Operating Range Constraints

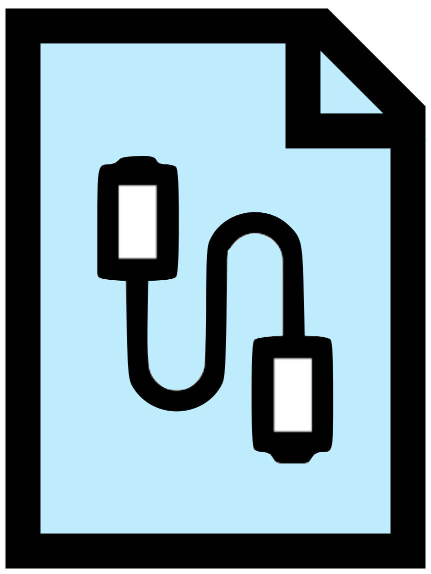


given a connection:

→ ensure signal is scaled equally on both sides of connection

CONNECTION CONSTRAINTS

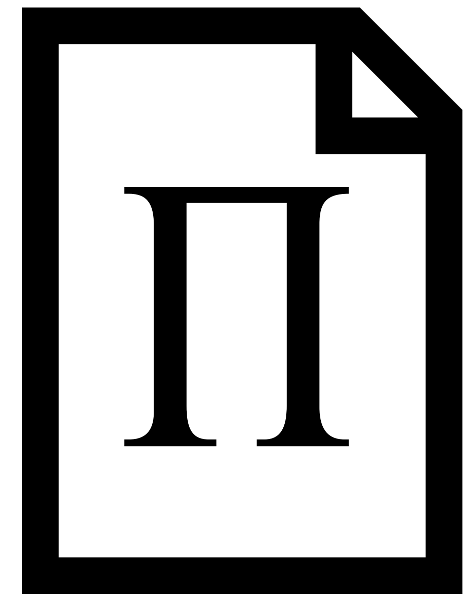




$$a_1 = a_6 \qquad a_2 = a_7 \qquad a_3 = a_8$$

GEOMETRIC PROGRAM GENERATION

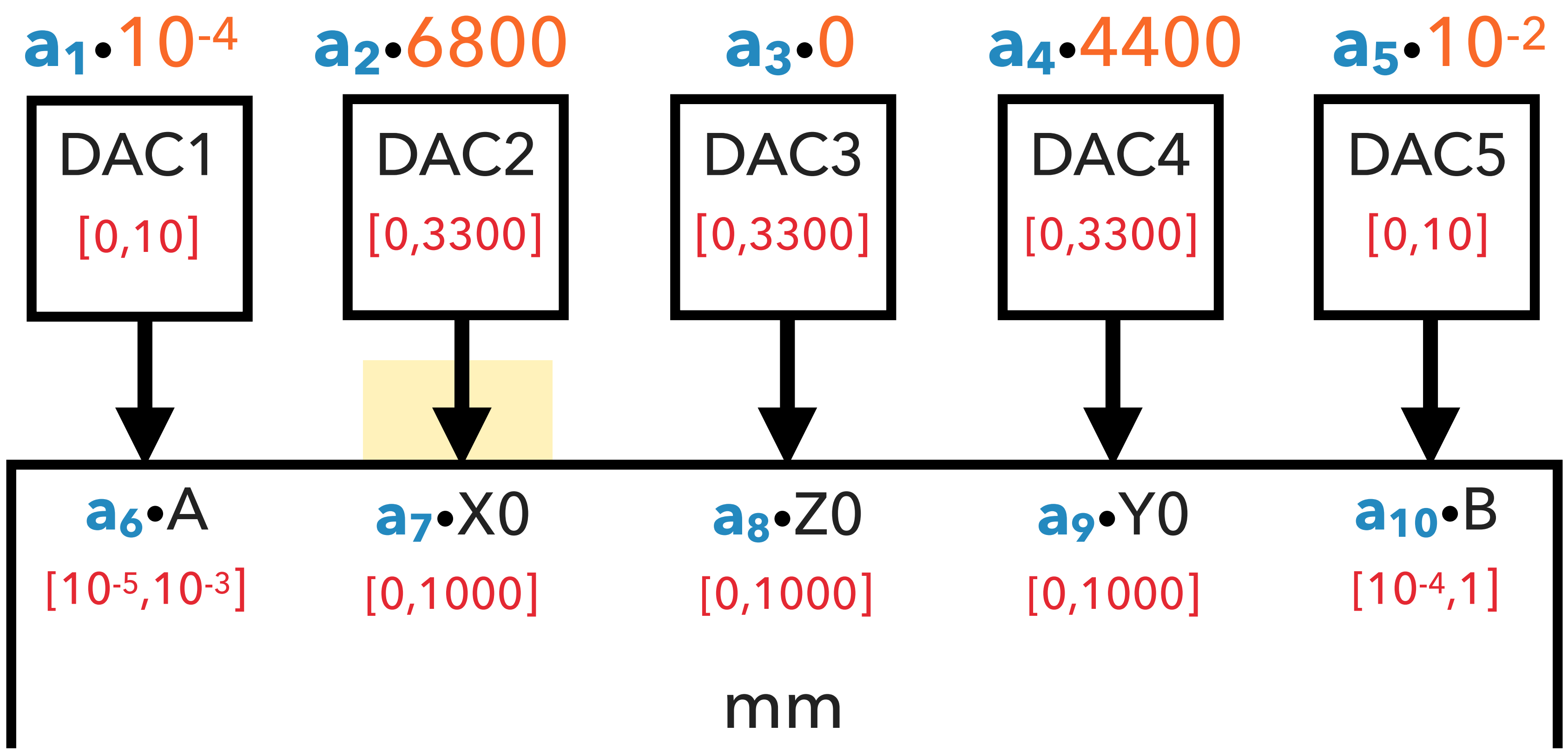
- Maximize τ subject to:
  - Factor Constraints
  - Sampling Constraints
  - Connection Constraints
  - Operating Range Constraints

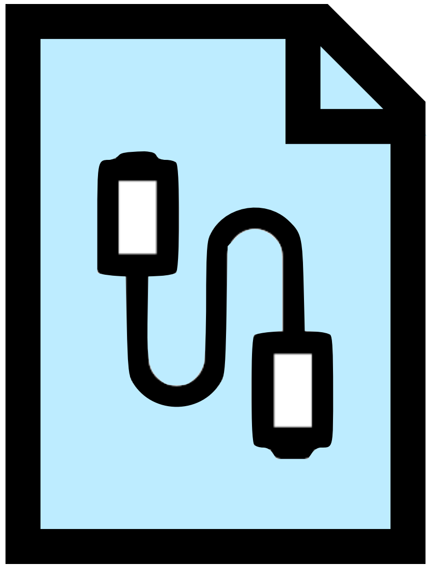


given an input/output port:

→ ensure signal stays within the operating range of the port

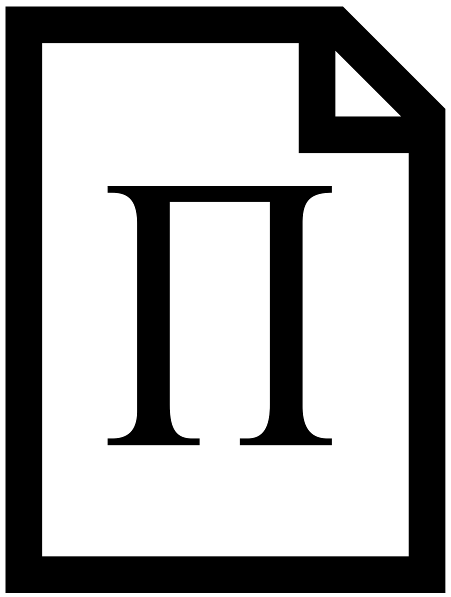
OPERATING RANGE RANGE CONSTRAINTS





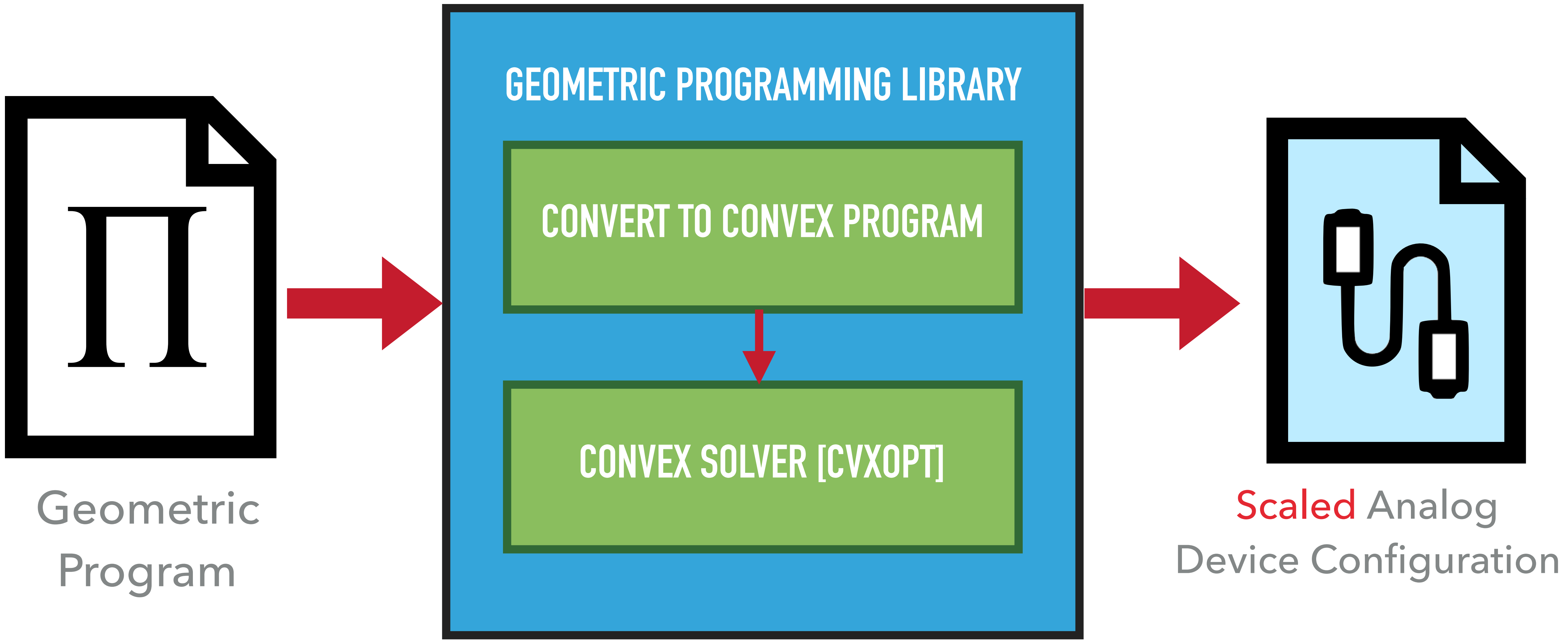
$$0 \leq a_7 \cdot 6800 \leq 1000$$

GEOMETRIC PROGRAM GENERATION

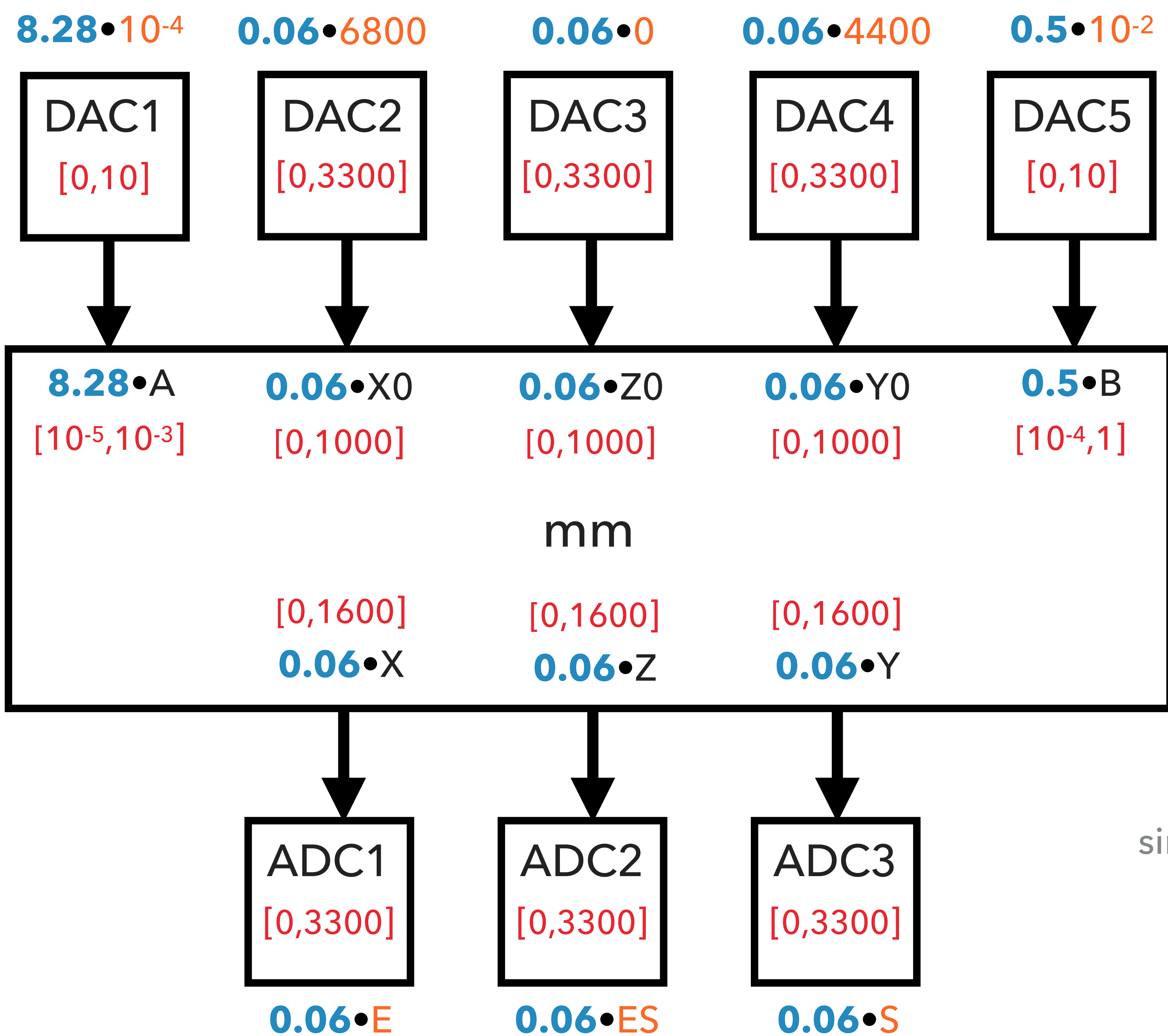


- Maximize τ subject to:
  - Factor Constraints
  - Sampling Constraints
  - Connection Constraints
  - Operating Range Constraints

GEOMETRIC PROGRAM GENERATION



SCALED ANALOG DEVICE CONFIGURATION



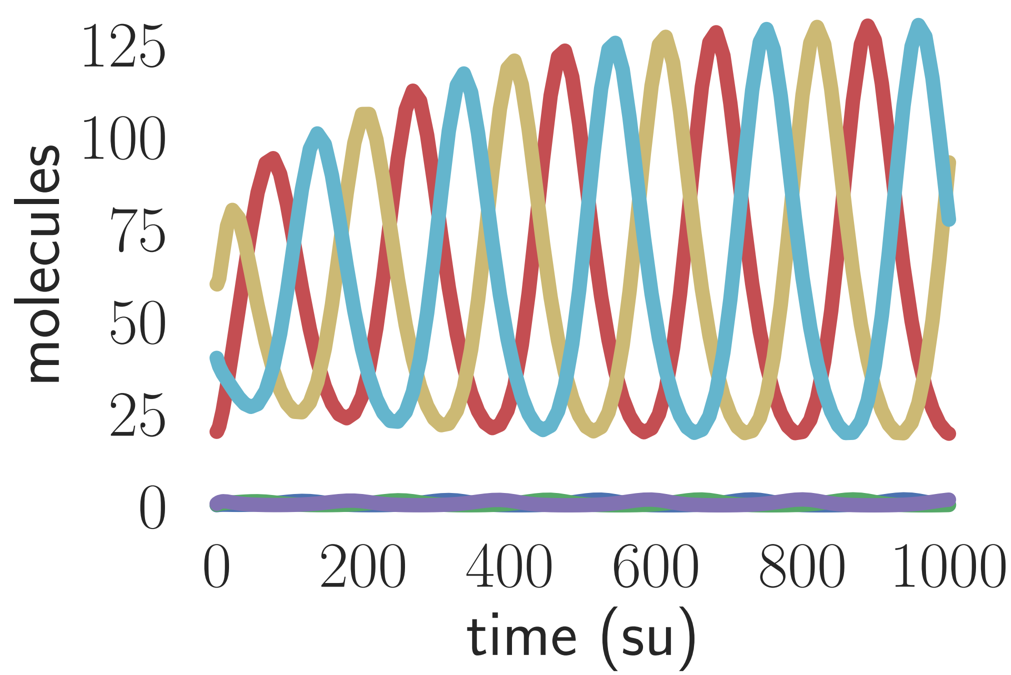
simulation speed

0.5

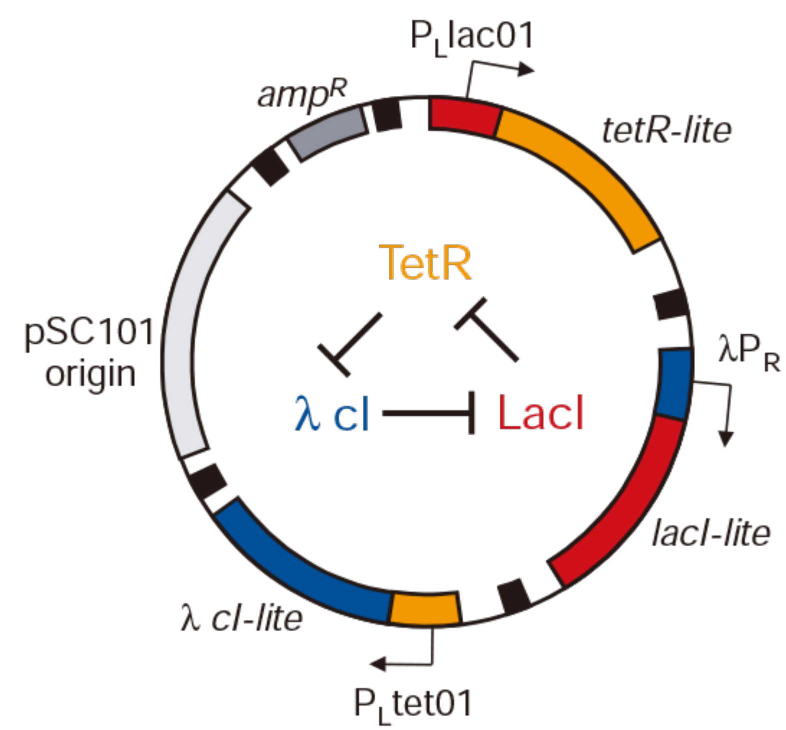
CASE STUDY: REPRISSILATOR

gene network that generates oscillations

“synthetic genetic clock”



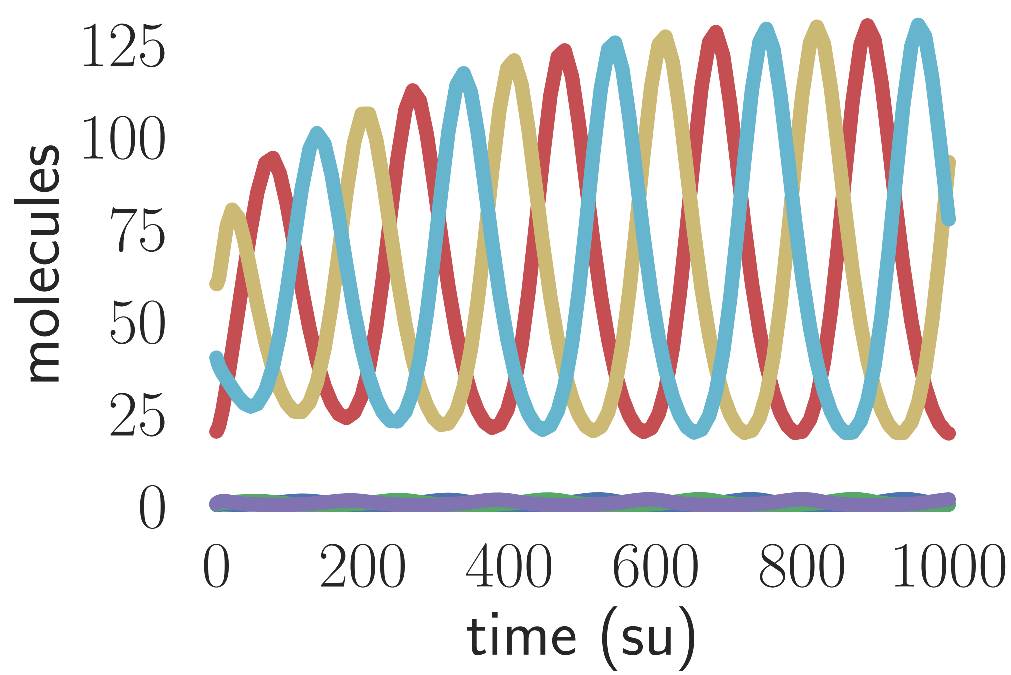
reference simulation



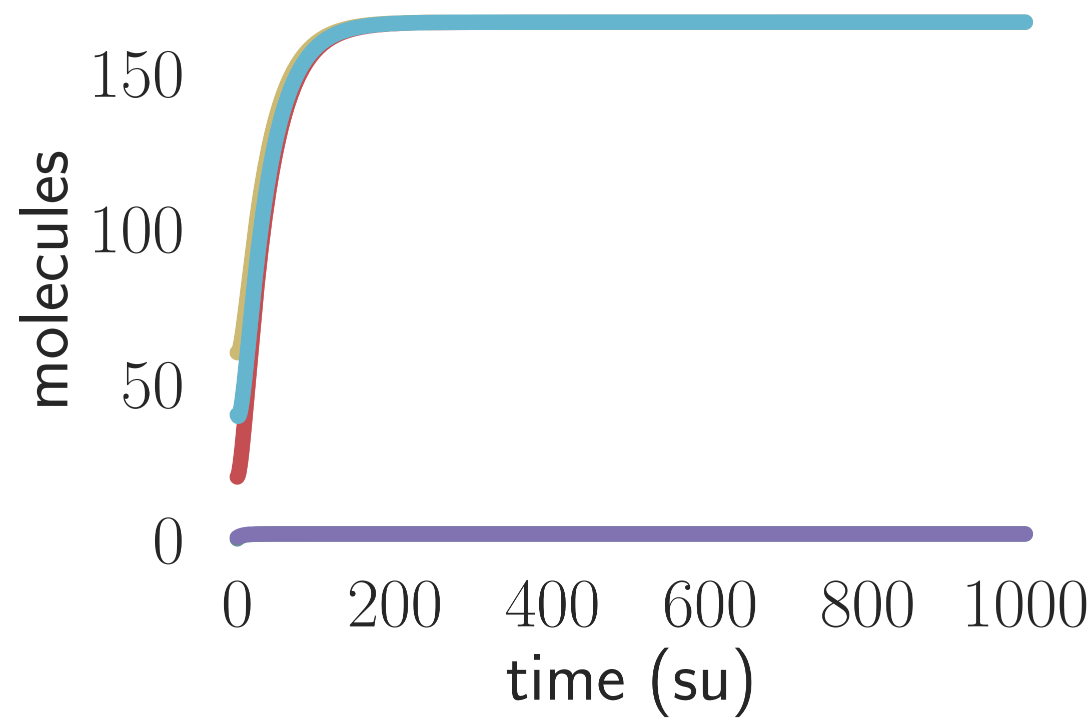
CASE STUDY: REPRISSILATOR

simulation using unscaled configuration

saturates, loses oscillations



reference simulation

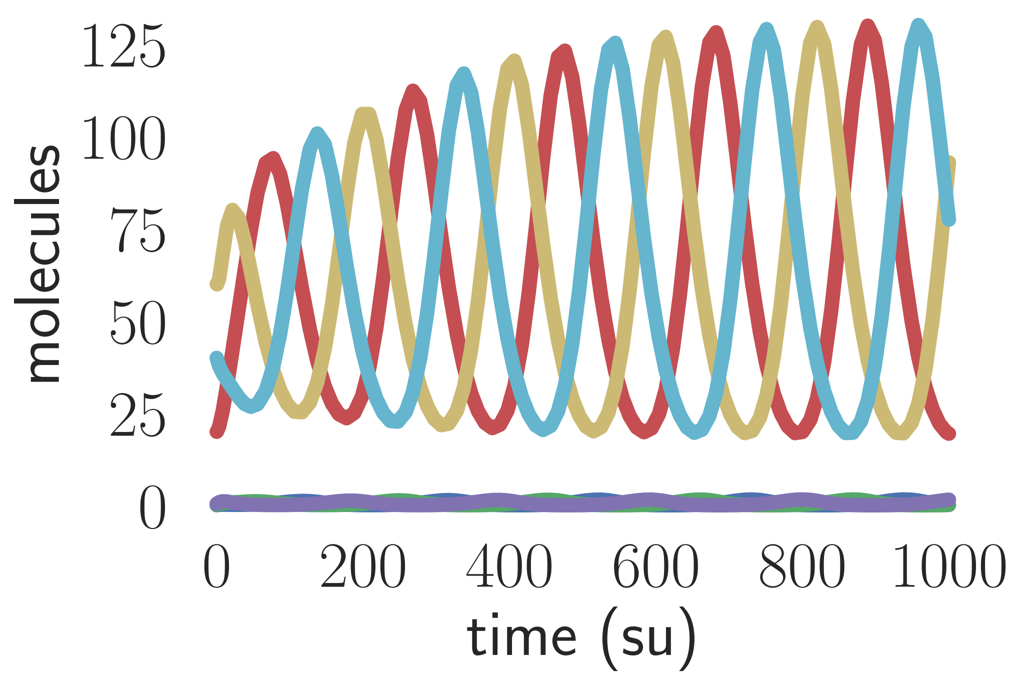


simulation **without jaunt**

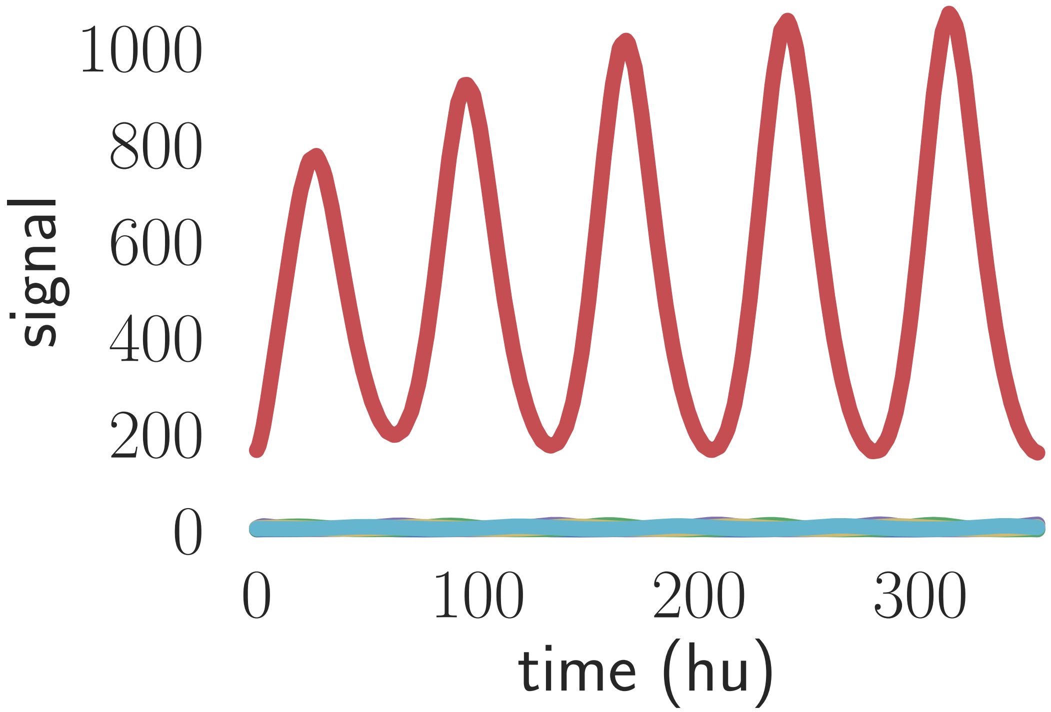
CASE STUDY: REPRISSILATOR

simulation using **scaled** configuration

executes 2.839x faster than unscaled configuration



Reference Simulation

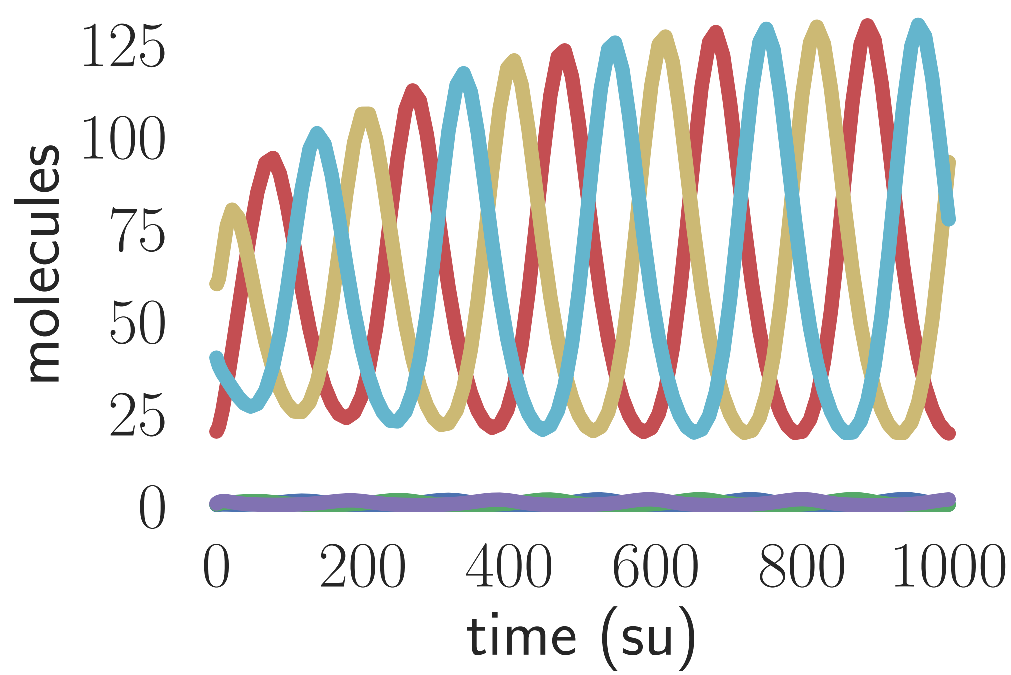


simulation **with jaunt before recovery**

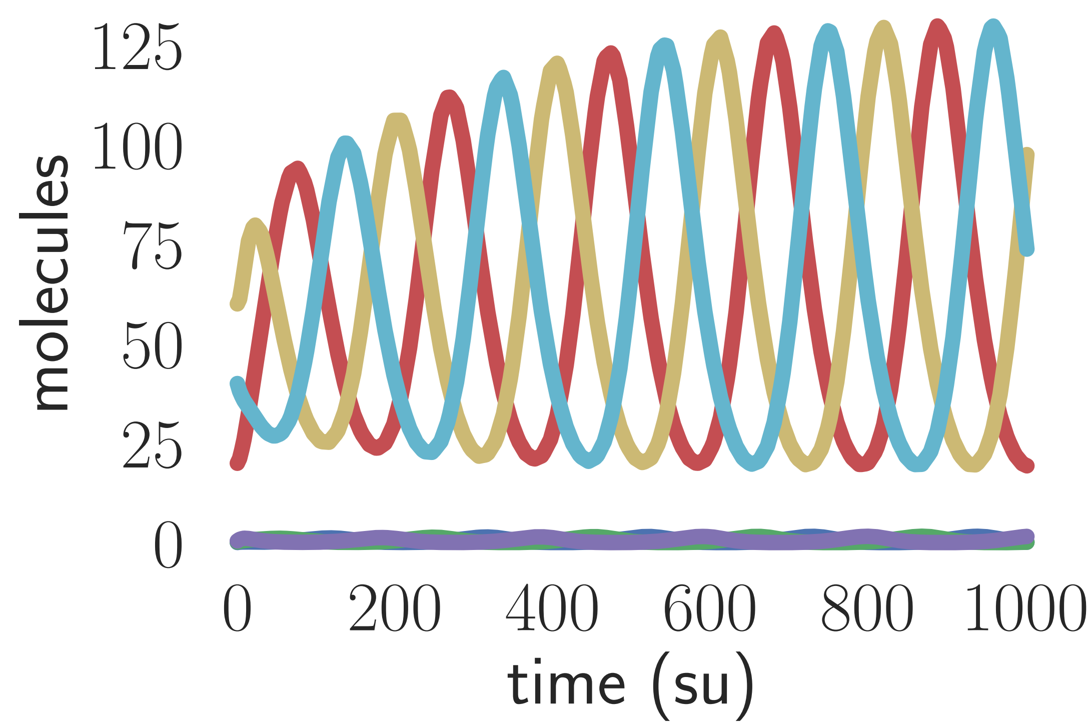
CASE STUDY: REPRISSILATOR

simulation using **scaled** configuration

scaling samples and time recovers original simulation



Reference Simulation



simulation **with jaunt after recovery**

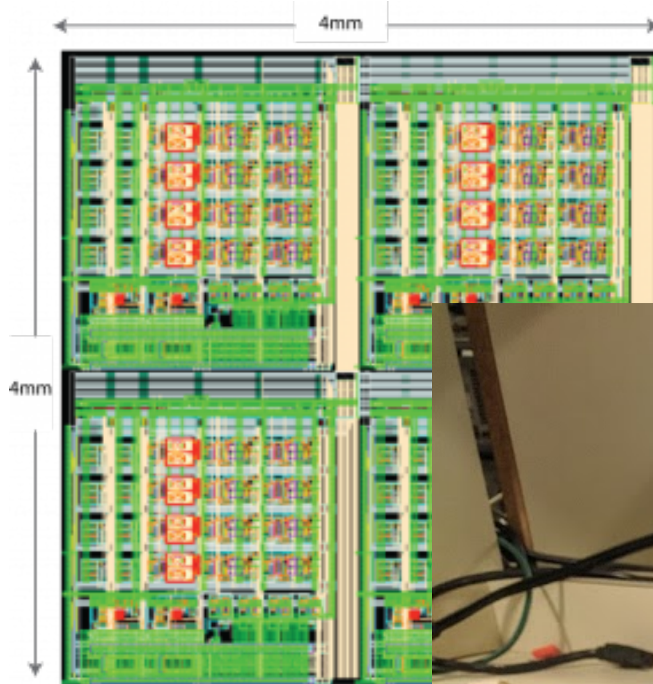
TALK OUTLINE

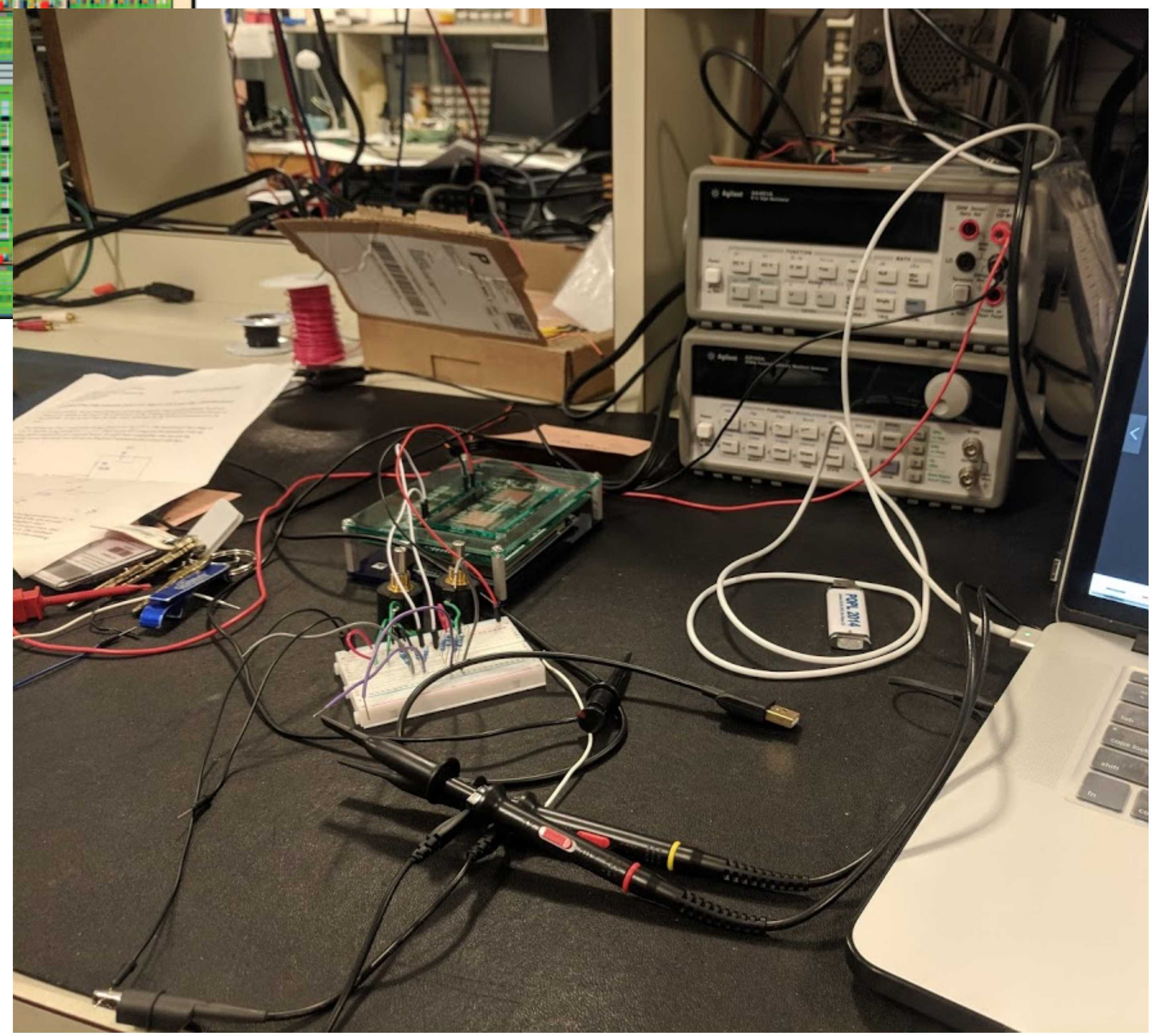
- ~~Background: overview of compilation problem~~
- ~~Arco Compiler[1]: automatically configure analog devices to simulate dynamical systems.~~
- ~~Jaunt Solver[2]: automatically scales dynamical systems to execute analog hardware with operating range constraints.~~
- Closing Remarks

1. Configuration Synthesis for Programmable Analog Devices with Arco. Sara Achour, Rahul Sarpeshkar and Martin Rinard. June 2016. PLDI 2016.
2. Time Dilation and Contraction for Programmable Analog Devices with Jaunt. Sara Achour and Martin Rinard. December 2017. ASPLOS 2018.

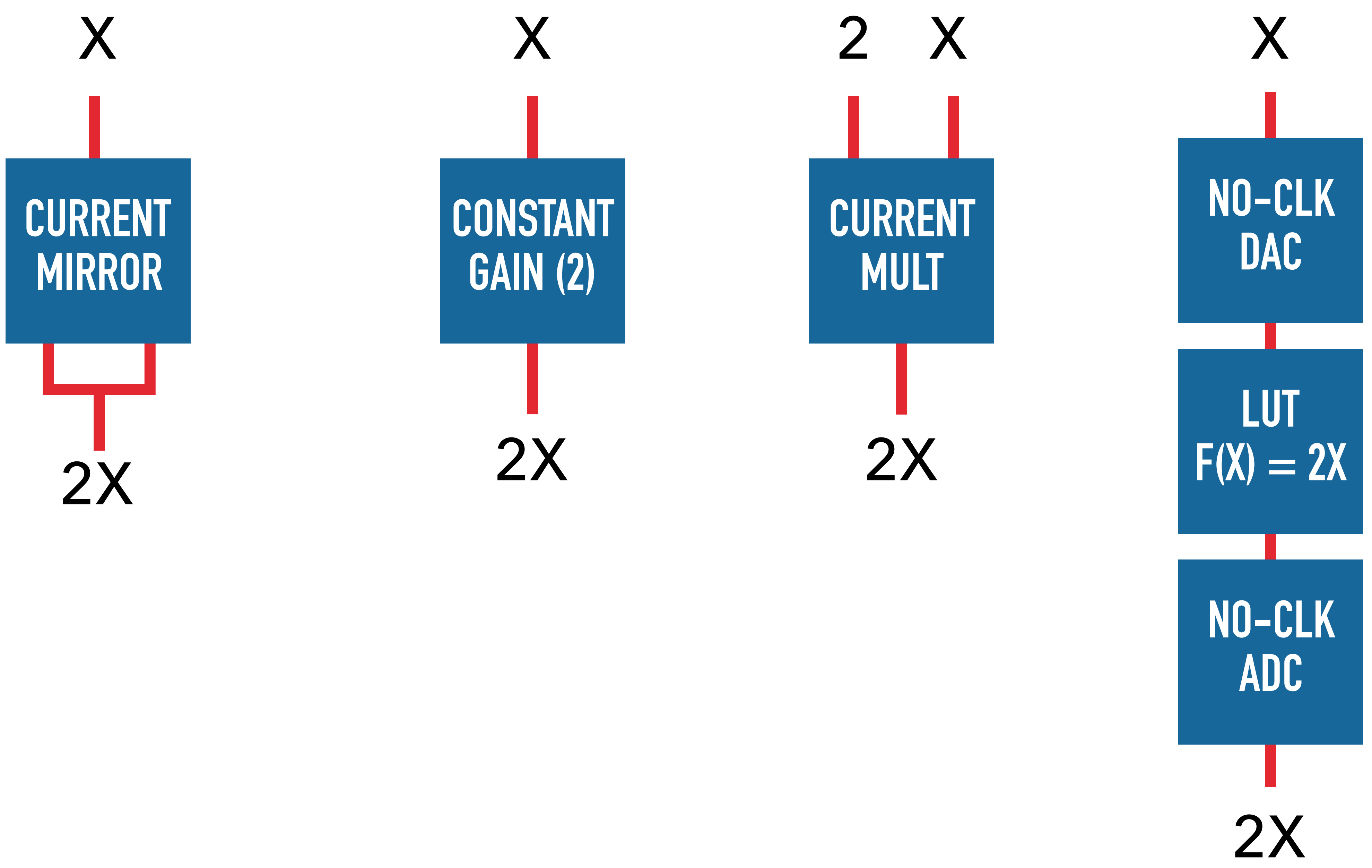
WHAT NEXT?

SENDYNE HYBRID DIGITAL-ANALOG COMPUTATION CHIP



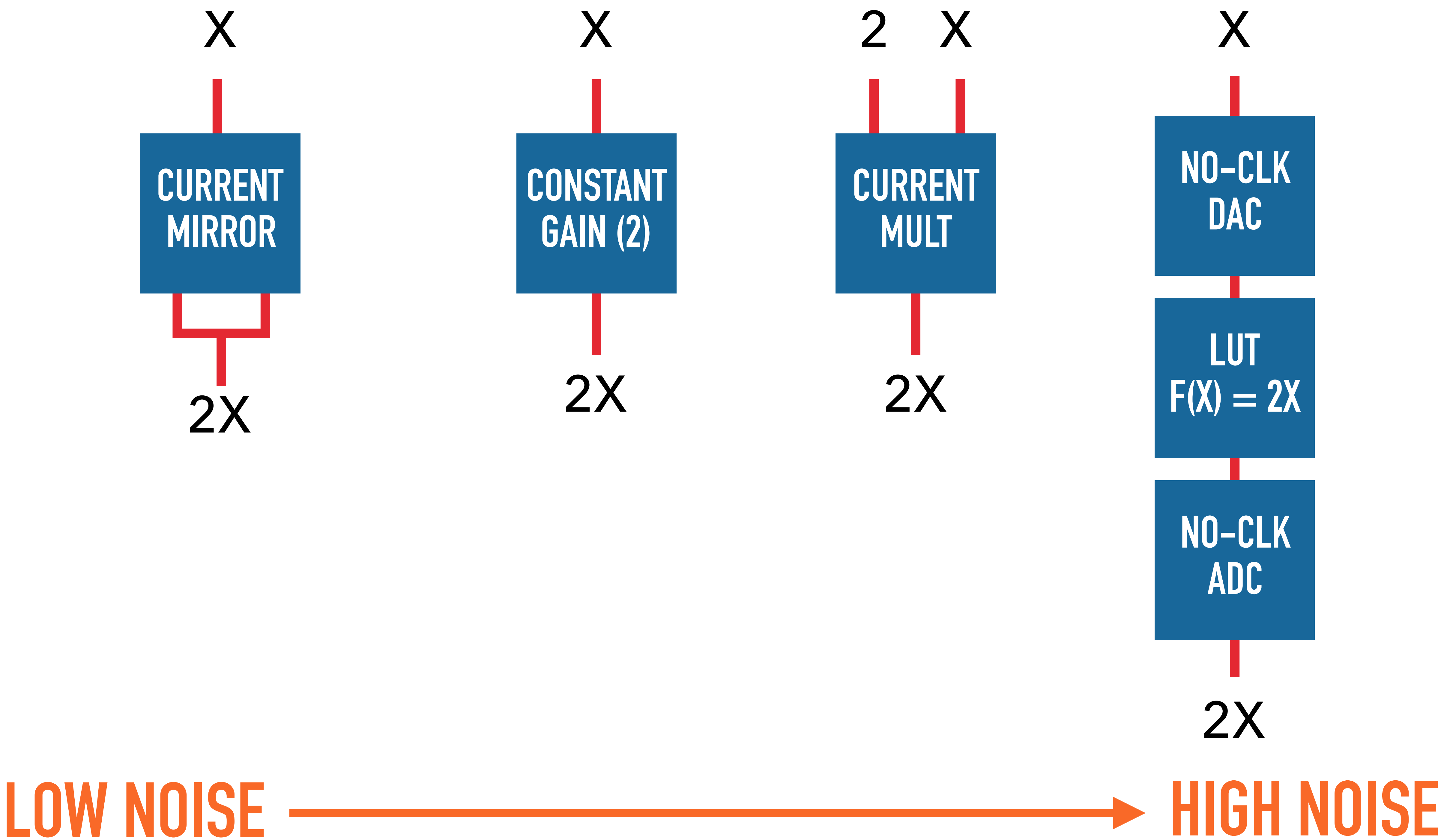


WHAT IS THE DIFFERENCE BETWEEN THESE CIRCUITS?



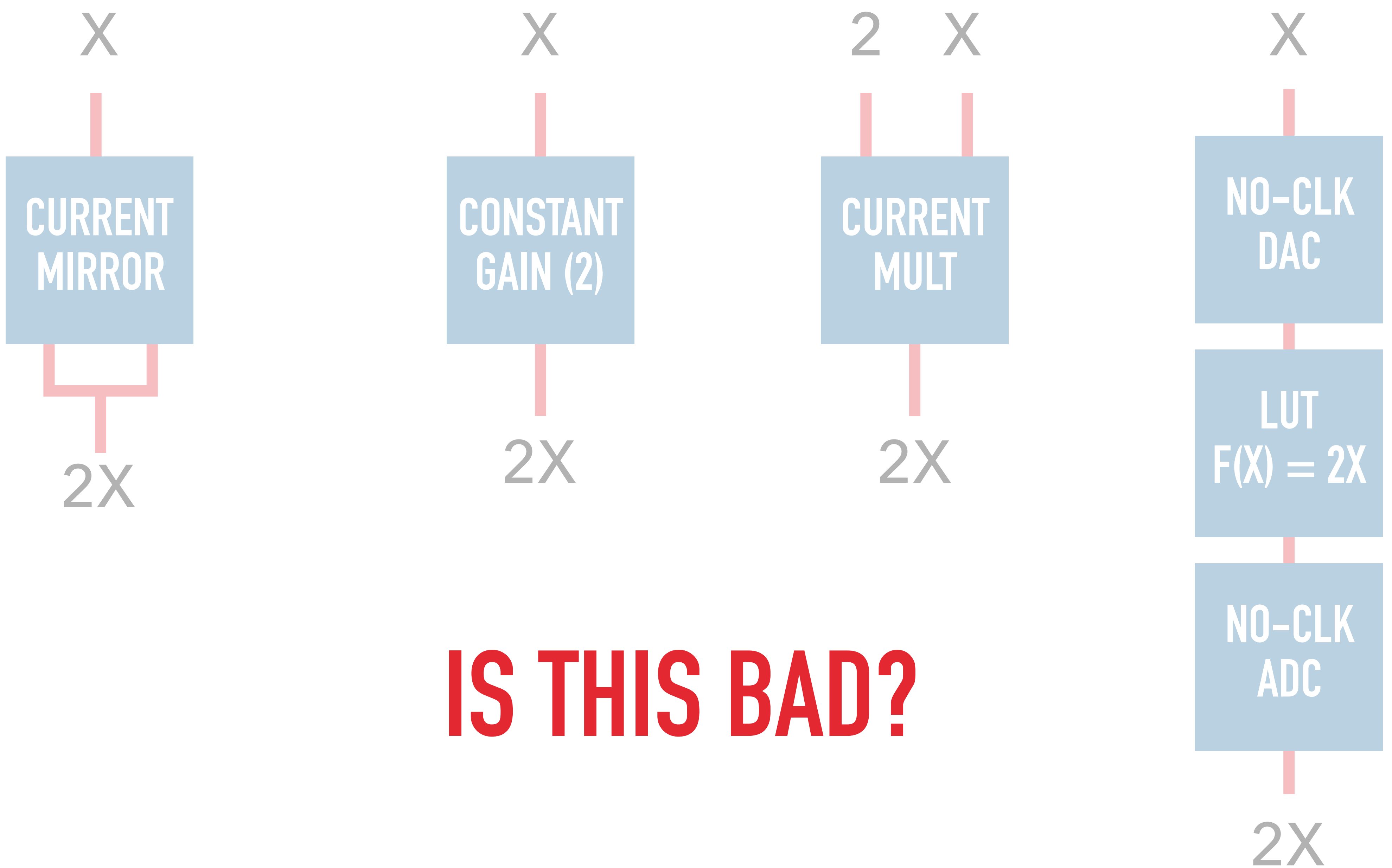
WHAT IS THE DIFFERENCE BETWEEN THESE CIRCUITS?

NOISE BEHAVIOR!



WHAT IS THE DIFFERENCE BETWEEN THESE CIRCUITS?

NOISE BEHAVIOR!



IS THIS BAD?

IS THIS BAD?

inherent variance in physical systems

SDES

other stochastic processes

circuit noise = stochastic behavior

IS THIS BAD?

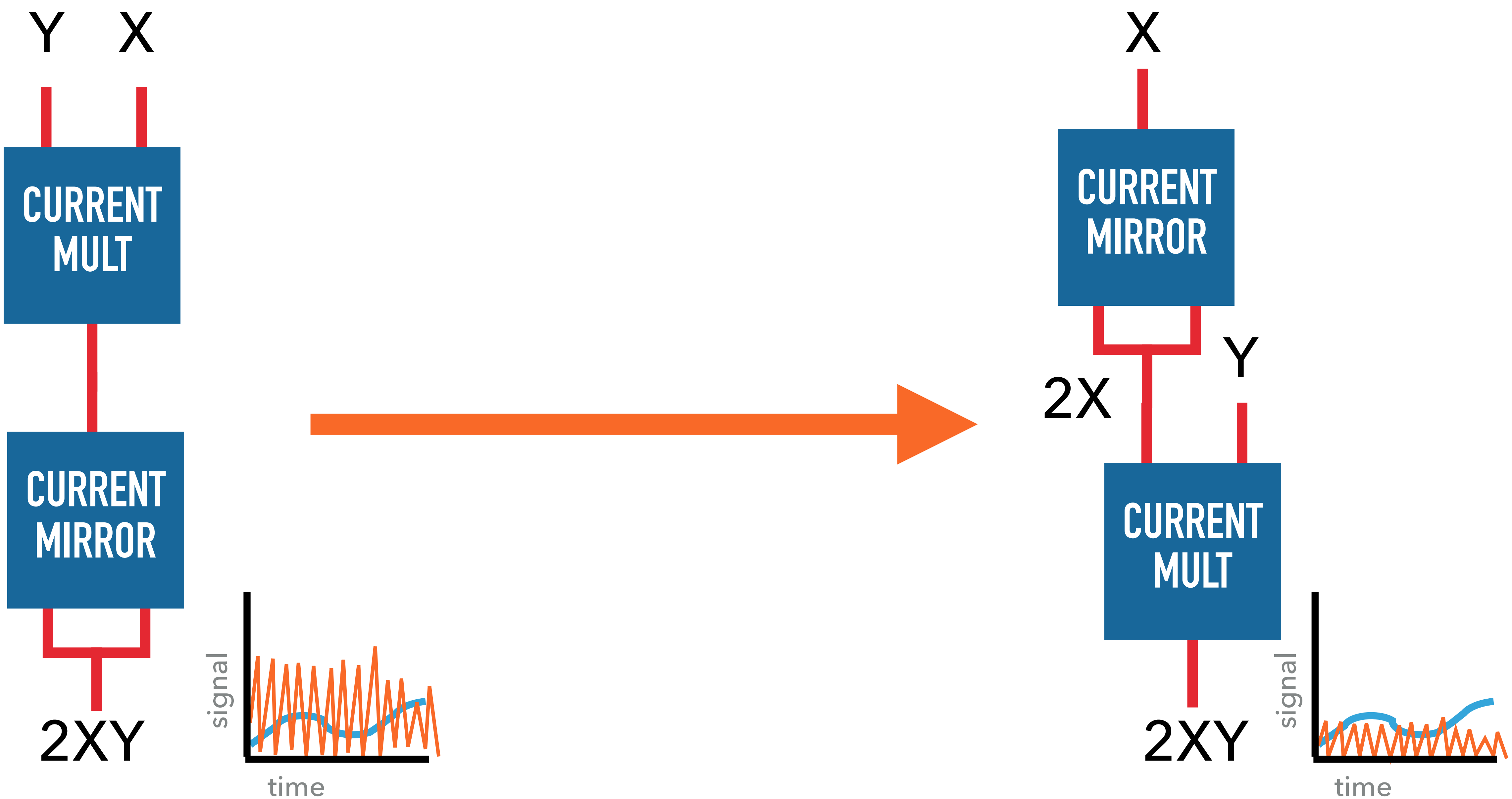
uncertainty in modeling physical systems

unmodeled dynamics

empirically derived models

circuit noise <= uncertainty

TECHNIQUES FOR MANIPULATING NOISE



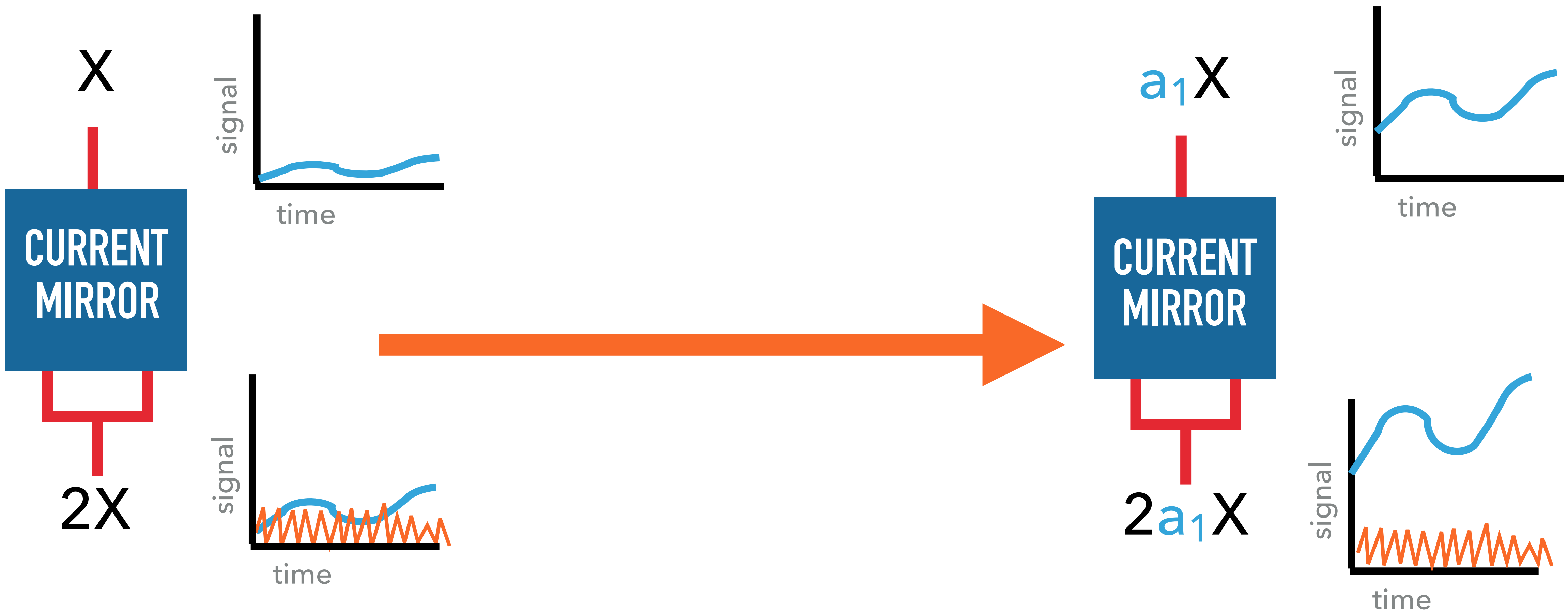
Option 1: Rearrange circuit to reduce noise

[noise-aware circuit generation]

Signal

Noise

TECHNIQUES FOR MANIPULATING NOISE



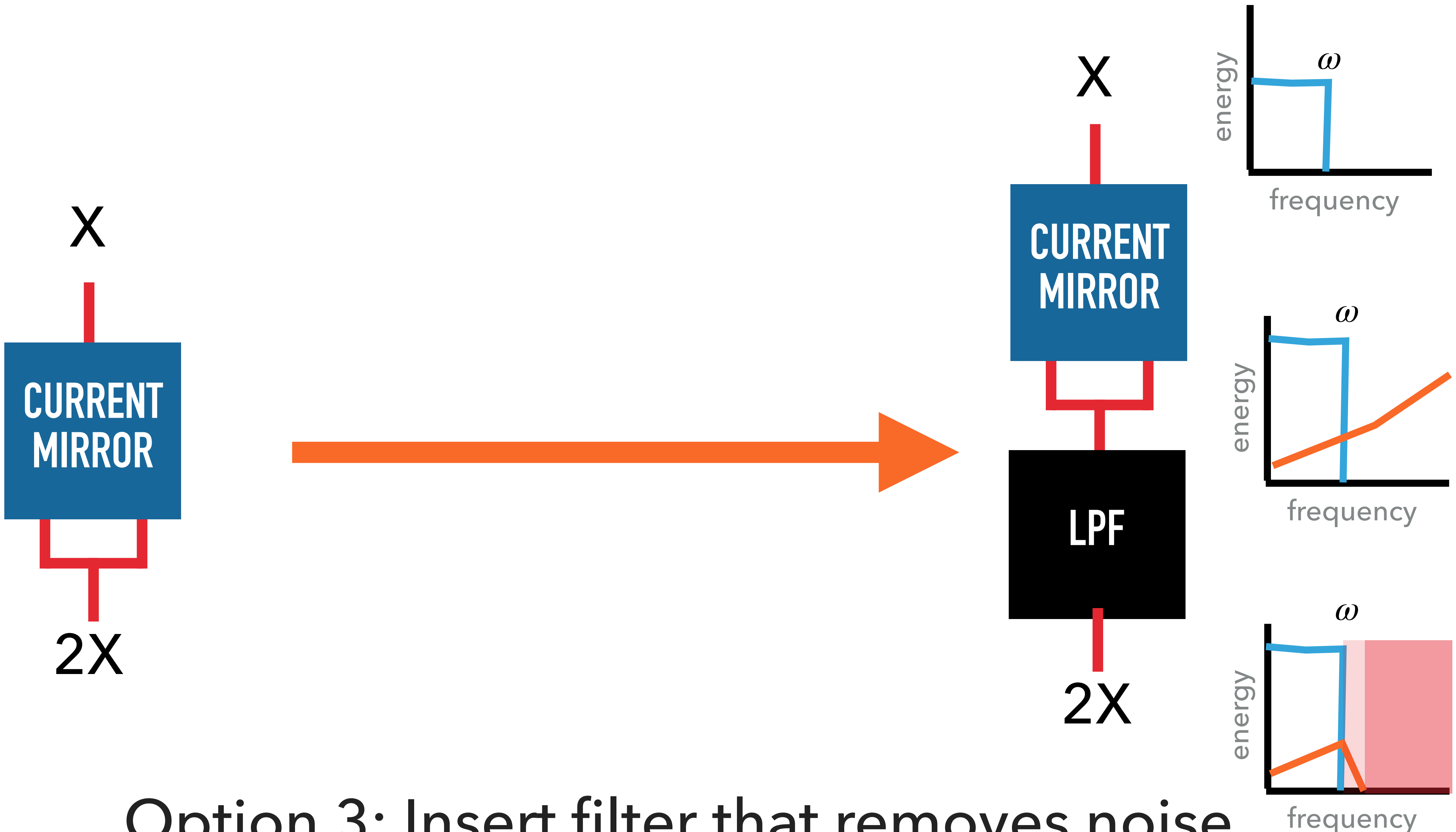
<u>Option 2</u>: Increase dynamic range of X

[noise-aware parameter scaling]

Signal

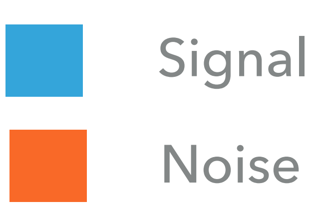
Noise

TECHNIQUES FOR MANIPULATING NOISE



Option 3: Insert filter that removes noise

[automated filter configuration]



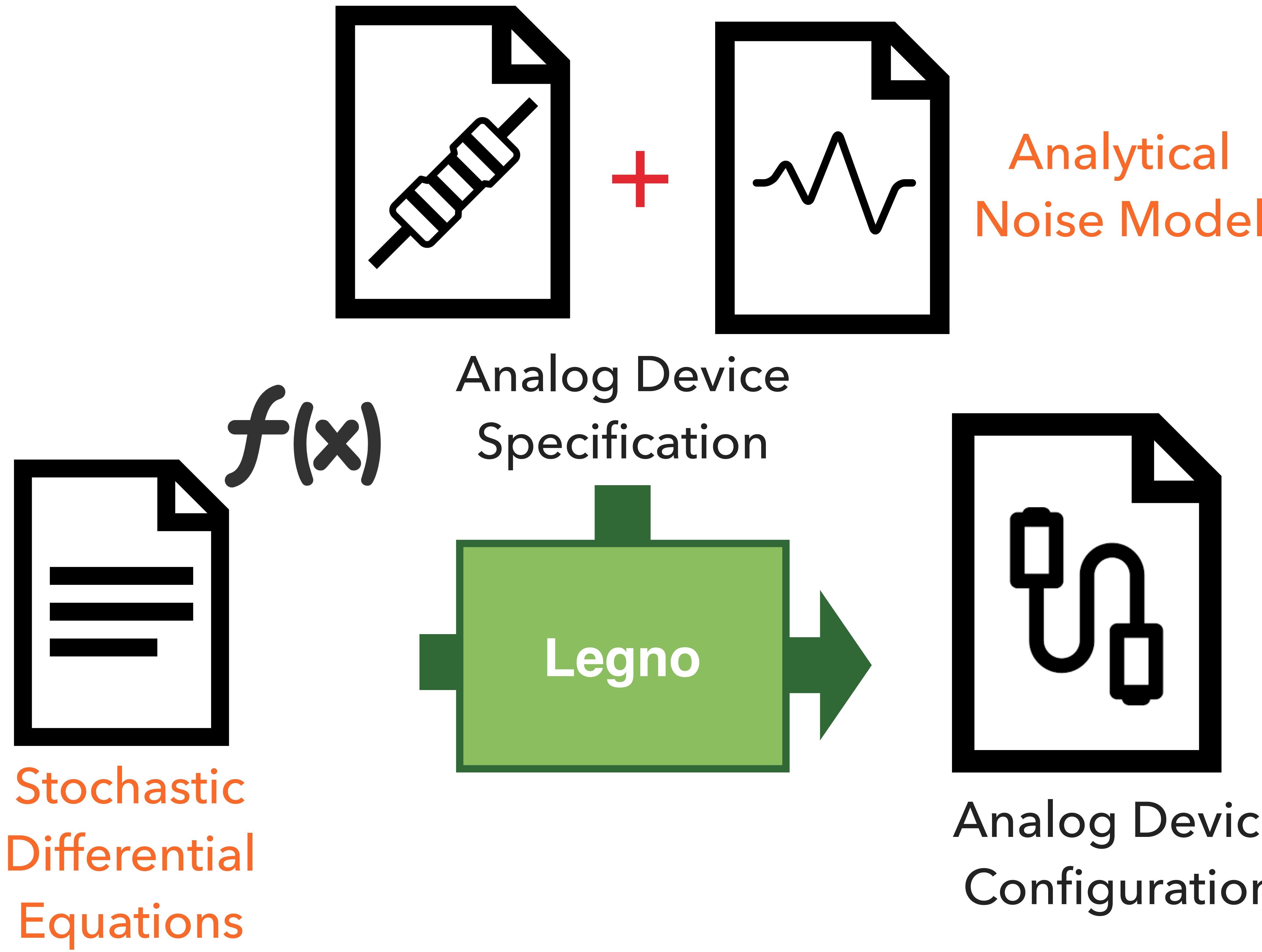
WHAT NEXT?

Legno: noise-aware configuration generation

noise-aware scaling transforms

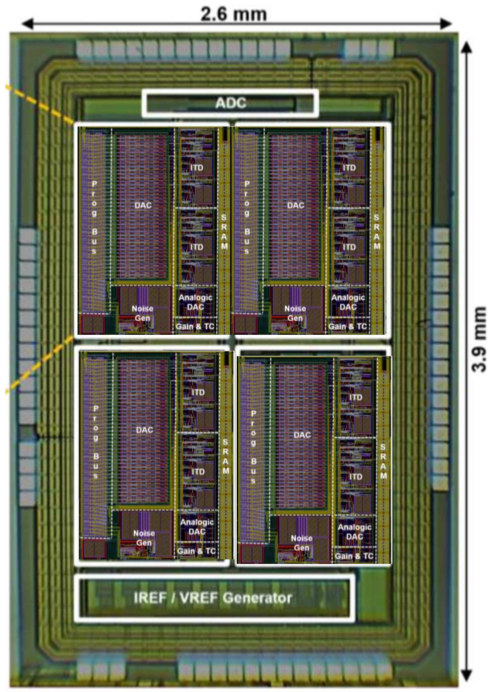
automated filter generation

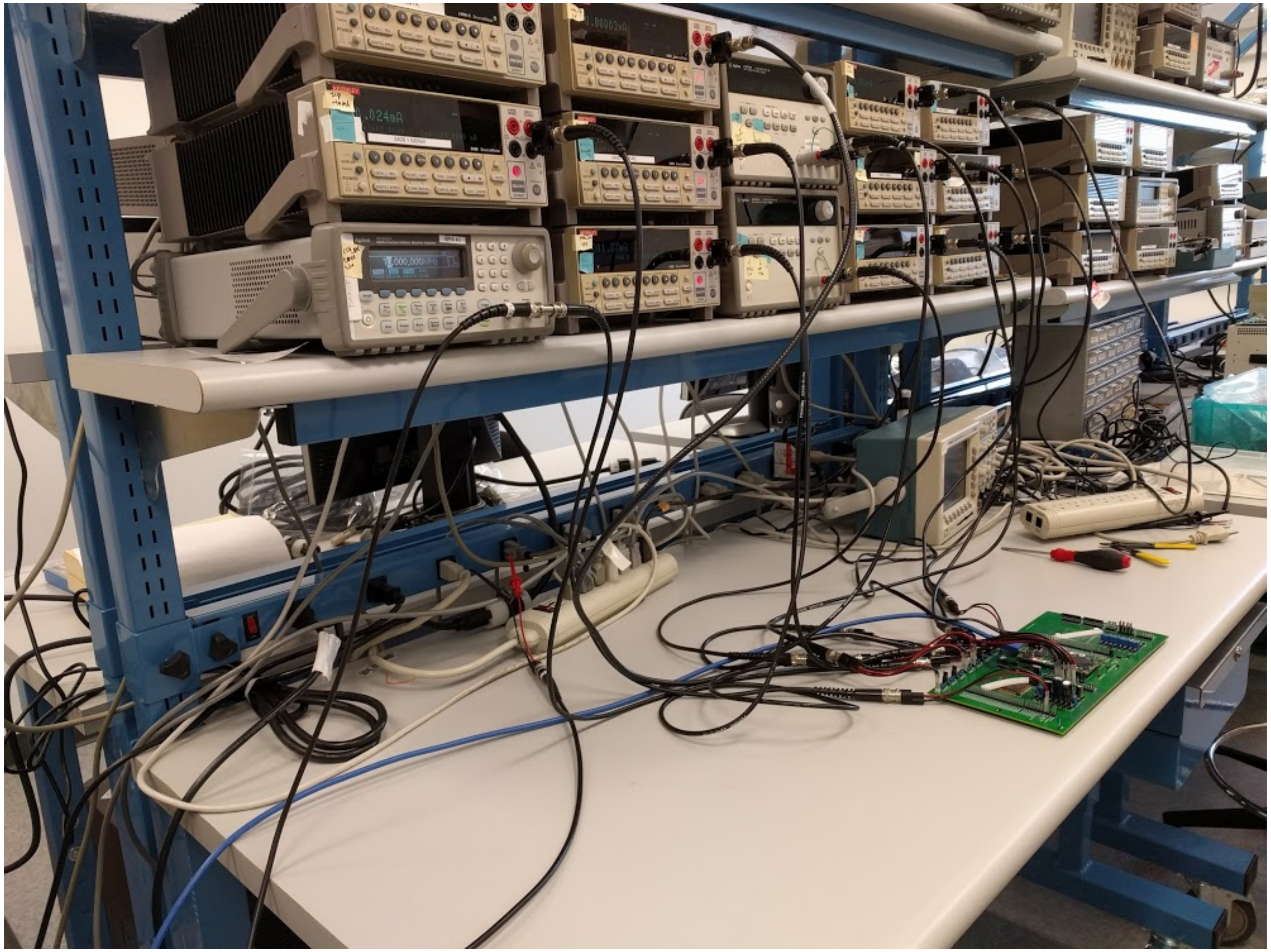
ranked configuration generation


Analytical
Noise Model
Analog Device
Specification
f(x)
Legno
Stochastic
Differential
Equations
Analog Device
Configuration

BACKUP SLIDES

SARPSHKAR GROUP PROTEIN CHIP





The Search Algorithm

- **Frontier (F):** Tableau configurations to explore
- **choose**: chooses the tableau **t** in **F** to explore.
- **select**: chooses the set of transitions to apply to **t**

Algorithm:

F = initial tableau

while **F**, choose **t** in **F**:

if **t** is terminal return **Z**

otherwise:

select **T:** set of t' where **t**→t'

remove **t** from **F**, add **T** to **F**

Search Optimizations

- **Search Heuristics**
  - choose lowest complexity tableau configuration.
  - select a simple goal, and prioritize transitions that solve it.

Algorithm:

F = initial tableau
while **F**, *choose* **t** in **F**:
  if **t** is terminal return **Z**
  otherwise:
    select **T:** set of t' where **t**→t'
    remove **t** from **F**, add **T** to **F**

Search Optimizations

- **<u>Component Aggregation</u>**: aggregate instances of the same component
    - **Pro**: smaller search space
    - **Con**: instance constraints must be handled separately
- **<u>Partial Configuration Caching</u>**
- **<u>Compact Search Tree Data Structure</u>**

1. Transitions Over Tableau

$$\frac{\overset{\textbf{UNIFY}}{\mathrm{r} \in \overline{\mathrm{R}} \cup \dot{\mathrm{R}} \qquad \widetilde{\mathrm{r}} \in \widetilde{\mathrm{R}} \qquad \textsf{unify}(\mathrm{r}, \widetilde{\mathrm{r}}, \overline{\mathrm{R}}, \dot{\mathrm{R}}, \widetilde{\mathrm{R}}) = \langle \overline{\mathrm{R}}', \dot{\mathrm{R}}', \widetilde{\mathrm{R}}' \rangle}}{\langle \overline{\mathrm{R}}, \dot{\mathrm{R}}, \mathrm{W}, \widetilde{\mathrm{R}}, \mathrm{Z} \rangle \to \langle \overline{\mathrm{R}}', \dot{\mathrm{R}}', \mathrm{W}, \widetilde{\mathrm{R}}', \mathrm{Z} \rangle}$$

$$\left\langle \quad \overline{R} \qquad \dot{R} \qquad W \qquad \tilde{R} \qquad Z \quad \right\rangle$$

$W_D = U_I$	$\langle B, U \rangle$	$S = 2 \cdot T$
$Z_I = X_D$	$\langle C, U \rangle$	
$B_I = 2 \cdot A_I$	$\langle Z, A \rangle$	
$C_I = \frac{A_I}{2}$		
$F_I = \frac{E_I}{4}$		

1. Transitions Over Tableau

CONNECT

$$\frac{\widetilde{\mathrm{r}} : \overline{\mathtt{i}}_{\mathrm{q}} = \overline{\mathtt{o}}_{\mathrm{q}} \in \widetilde{\mathrm{R}} \qquad \mathrm{w} : \langle \overline{\mathtt{o}}, \overline{\mathtt{i}} \rangle \in \mathrm{W}}{\langle \overline{\mathrm{R}}, \dot{\mathrm{R}}, \mathrm{W}, \widetilde{\mathrm{R}}, \mathrm{Z} \rangle \rightarrow \langle \overline{\mathrm{R}}, \dot{\mathrm{R}}, \mathrm{W} - \{\mathrm{w}\}, \widetilde{\mathrm{R}} - \{\widetilde{\mathrm{r}}\}, \mathrm{Z} \cup \{\overline{\mathtt{o}} \bullet\!\!-\!\!\bullet \overline{\mathtt{i}}\} \rangle}$$

$$\langle \quad \overline{R} \quad \dot{R} \quad W \quad \tilde{R} \quad Z \quad \rangle$$

$\overline{R}$	$\dot{R}$	W	$\tilde{R}$
$W_D = U_I$	$C_I = \frac{T}{2}$	$\langle B, U \rangle$	$S \mapsto B_I$
$F_I = \frac{E_I}{4}$	$B_I = S$	$\langle C, U \rangle$	$A_I = Z_I$
	$B_I = 2 \cdot T$	$\langle Z, A \rangle$	$X_D = T$
	$Z_I = T$		

1. Transitions Over Tableau

INPUT-VAR-MAP

$$\frac{\widetilde{\mathrm{r}} : \bar{\mathtt{i}}_{\mathrm{d}} = \widehat{i} \in \widetilde{\mathrm{R}} \qquad \widehat{i} \in \widehat{I} \qquad \bar{\mathtt{i}} \mathrel{@} \bar{\mathtt{c}} \qquad \bar{\mathtt{c}} \in \overline{\mathtt{IC}}}{\langle \overline{\mathrm{R}}, \dot{\mathrm{R}}, \mathrm{W}, \widetilde{\mathrm{R}}, \mathrm{Z} \rangle \to \langle \overline{\mathrm{R}}, \dot{\mathrm{R}}, \mathrm{W}, \widetilde{\mathrm{R}} - \{\widetilde{\mathrm{r}}\}, \mathrm{Z} \cup \{\widehat{i} \mapsto \bar{\mathtt{i}}_{\mathrm{d}}\} \rangle}$$

$$\langle \quad \overline{R} \qquad \dot{R} \qquad W \qquad \tilde{R} \qquad Z \quad \rangle$$

$\overline{R}$	$\dot{R}$	W	$\tilde{R}$	Z
$W_D = U_I$	$C_I = \frac{T}{2}$	$\langle B, U \rangle$	$S \mapsto B_I$	$Z \bullet\!\!-\!\!\bullet A$
$F_I = \frac{E_I}{4}$	$B_I = S$	$\langle C, U \rangle$	$X_D = T$	$T \mapsto X_D$
	$B_I = 2 \cdot T$			
	$Z_I = T$			

Geometric Programming Problem

$$minimize\ s_{opt}$$

$$m_i \le 1, i = 1, \ldots, n$$

$$s_j = 1, j = 1, \ldots, m$$

monomial m_i

$$m_i = c_i \tau^{x_{i,\tau}} \prod_{p \in P} a_p^{x_{i,p}}$$

posynomial p_i

$$s_i = \sum_i c_i \tau^{x_{i,\tau}} \prod_{p \in P} a_p^{x_{i,p}}$$

Analog Hardware Components

selection of analog components from collaborators, textbooks and publications

Component	Quantity	Description	Relation
iin	25	current input	$Z_I = X_D$
vin	125	voltage input	$Z_V = X_D$
outi	10	current output	$Z_D = X_I$
vout	75	voltage output	$Z_D = X_V$
vgain	40	voltage gain	$O_V = (X_V \cdot Z_V)/(Y_V \cdot 25)$
iadd	30	current adder	$O_I = A_I + B_I + C_I + D_I$
vadd	35	voltage adder	$\partial O2_V/\partial t = 0.1(A_V + B_V - C_V - D_V \cdot O2_V)$
			$O1_V = 0.1(A_V + B_V - C_V - D_V)$
vtoi	30	voltage to current converter	$O_I = X_V/K_V$
itov	30	current to voltage converter	$O_I = K_V \cdot X_I$
ihill	8	hill function for activation/repression	$S_I = M_V(S_I/K_I)^{nv}/((S_I/K_I)^{nv} + 1)$
			$R_I = M_V/((S_I/K_I)^{nv} + 1)$
igenebind	8	gene binding	$O_I = M_I/(1 + K_I \cdot T_I)$
switch	15	genetic switch	$O_I = M_I/(S_I/K_I + 1)^{nv}$
mm	2	Michaelis-Menten dynamics	$X_V = Xt_V - XY_V$
			$Y_V = Yt_V - XY_V$
			$\partial XY_V/\partial t = K_I \cdot X_V \cdot Y_V - R_I \cdot XY_V$

G. Cowan, R. Melville, and Y. Tsividis. A VLSI analog computer/digital computer accelerator. *Solid-State Circuits, IEEE Journal of*, 41(1):42–53, Jan 2006. ISSN 0018-9200. doi: 10.1109/JSSC.

R. Daniel, S. S. Woo, L. Turicchia, and R. Sarpeshkar. Analog transistor models of bacterial genetic circuits. In *Biomedical Circuits and Systems Conference (BioCAS), 2011 IEEE*, IEEE, 2011.

R. Sarpeshkar. Ultra Low Power Bioelectronics: Funda- mentals, Biomedical Applications, and Bio-Inspired Systems. Cambridge University Press, 2010. ISBN 0521857279.

J. J. Y. Teo, S. S. Woo, and R. Sarpeshkar. Synthetic biology: A unifying view and review using analog circuits. *IEEE Trans. Biomed. Circuits and Systems*, 9(4):453–474, 2015.

S. S. Woo, J. Kim, and R. Sarpeshkar. A cytomorphic chip for quantitative modeling of fundamental bio-molecular circuits. *IEEE Trans. Biomed. Circuits and Systems*, 9(4):527–542, 2015.

Dynamical Systems Benchmarks

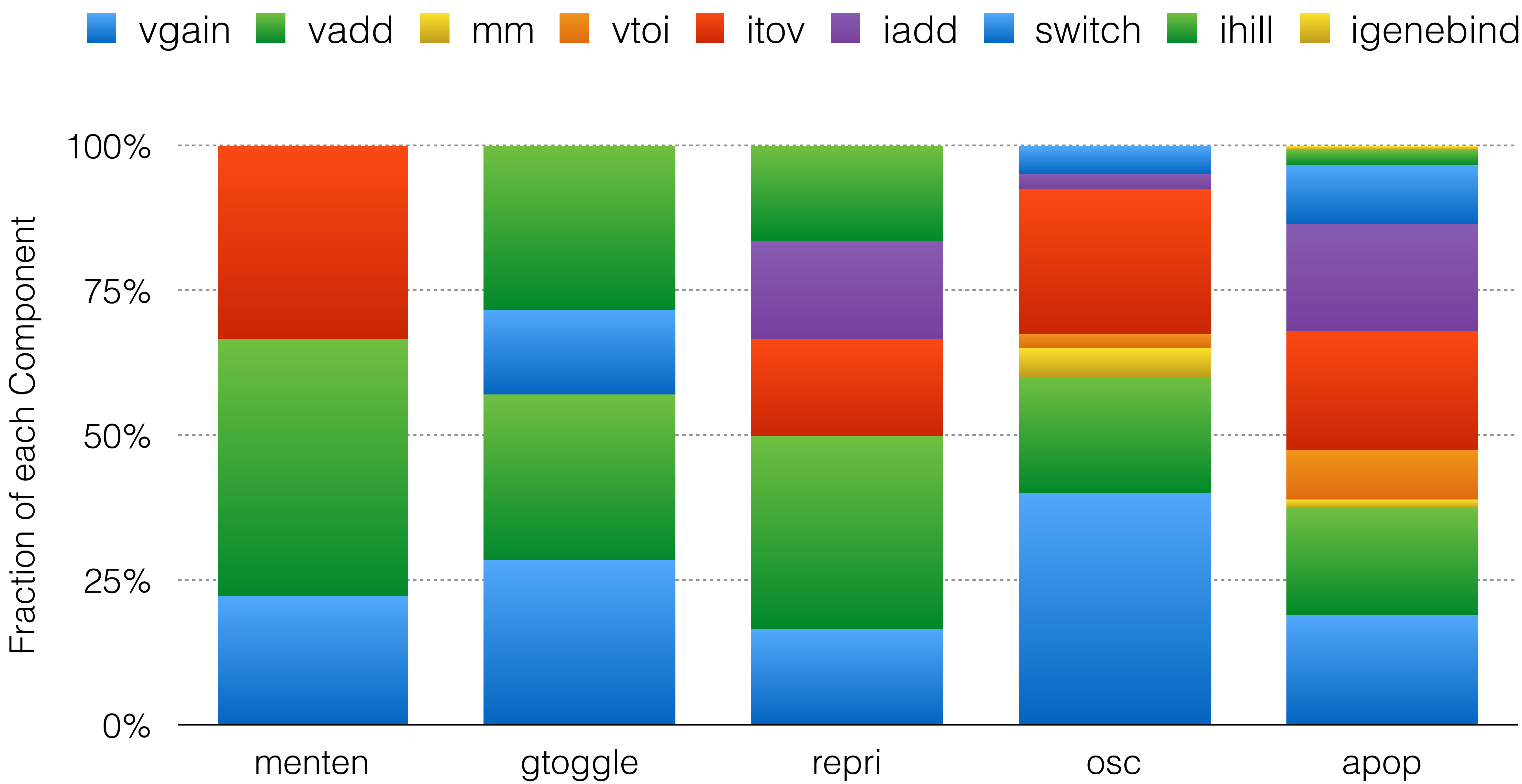
selection of published artifacts from well- cited computational biology papers from `Biomodels` database

Benchmark	Parameters	Functions	Differential Equations
menten	3	0	4
gentoggle	9	3	2
repr	7	3	6
osc	16	16	9
apop	87	48	27

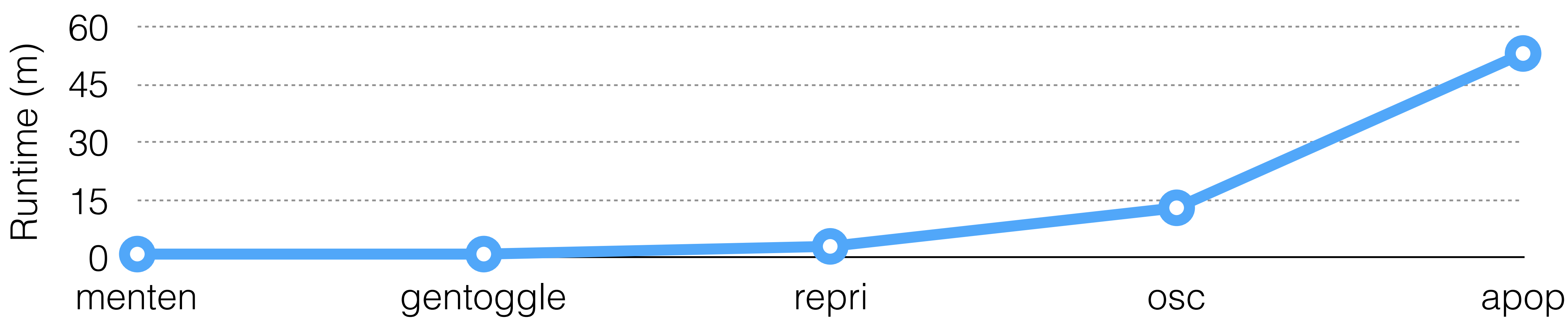
- **menten**: Michaelis-Menten equation reaction. D. R. F. PhD. Biochemistry (Lippincott Illustrated Reviews Series). LWW, 2013. ISBN 1451175620.
- **gentoggle**: genetic toggle switch in E.col. T. S. Gardner, C. R. Cantor, and J. J. Collins. Construction of a genetic toggle switch in escherichia coli. *Nature*, 403(6767): 339–342, 2000.
- **repri**: synthetic oscillatory network of transcriptional regulators. M. B. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338, 2000.
- **osc**: circadian oscillation utilizing activator / repressor. J. M. Vilar, H. Y. Kueh, N. Barkai, and S. Leibler. Mechanisms of noise-resistance in genetic oscillators. *Proceedings of the National Academy of Sciences*, 99(9):5988–5992, 2002
- **apop**: protein stress response. K. Erguler, M. Pieri, and C. Deltas. A mathematical model of the unfolded protein stress response reveals the decision mechanism for recovery, adaptation and apoptosis. *BMC systems biology*, 7(1):16, 2013.

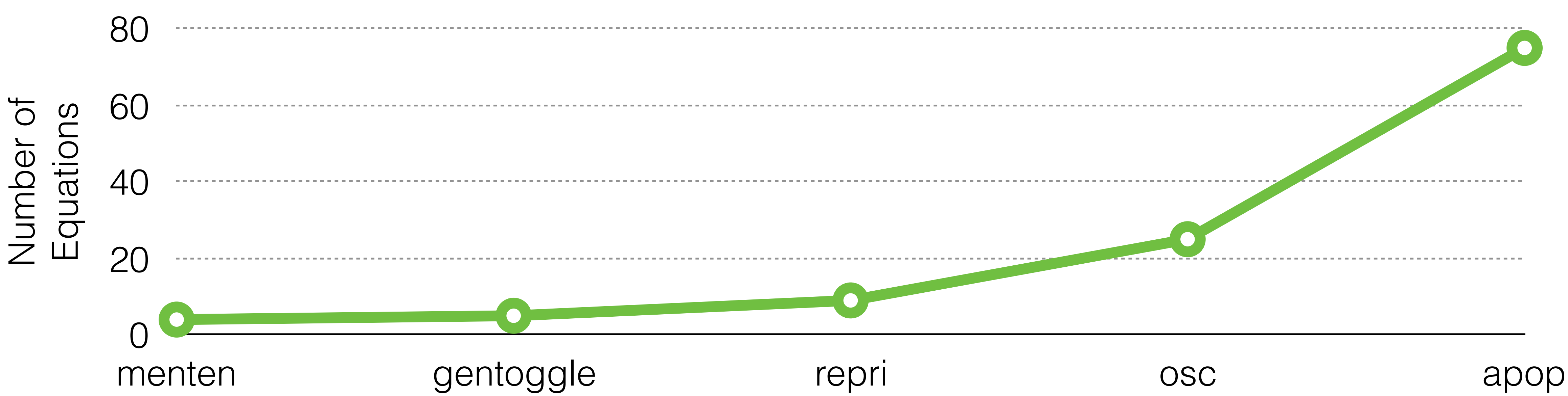
https://www.ebi.ac.uk/biomodels-main/

Components


vgain
vadd
mm
vtoi
itov
iadd
switch
ihill
igenebind
Fraction of each Component
100%
75%
50%
25%
0%
menten
gtoggle
repri
osc
apop

Arco Runtime





Geometric Programming Problem

$$minimize\ s_{opt}$$

$$m_i \leq 1, i = 1, \ldots, n$$

$$s_j = 1, j = 1, \ldots, m$$

monomial m_i

$$m_i = c_i \tau^{x_{i,\tau}} \prod_{p \in P} a_p^{x_{i,p}}$$

posynomial p_i

$$s_i = \sum_i c_i \tau^{x_{i,\tau}} \prod_{p \in P} a_p^{x_{i,p}}$$

Correctness+Speedup Results

benchmark	speedup	No Jaunt	Jaunt
smol	**0.50x**		✓
sconc	**1.00x**		✓
mmrxnp	77.48x		✓
repri	2.839x		✓
bont	**5.00x**	✓*	✓
epor	0.142x		✓
gtoggle	**0.1x**		✓

Simulation Speed Analysis

bmark	min τ	max τ	slow (hu)	fast (hu)	slow (s)	fast (s)
smol	0.010x	0.500x	1000 hu	20 hu	2s	0.004s
sconc	0.0001x	1.000x	25000 hu	25 hu	50s	0.05s
mmrxn	0.0001x	77.482x	$6 \cdot 10^6$ hu	77 hu	3h20m	0.155s
gtoggle	0.099x	0.100x	100 hu	100 hu	0.2s	0.2s
repri	0.0001x	2.839x	10^6 hu	352 hu	32m40s	0.704s
bont	0.0001x	5.000x	$25 \cdot 10^4$ hu	50 hu	8m20s	0.1s
epor	0.0001x	0.142x	10^6 hu	704 hu	3m20s	1.408s

Table 4. Jaunt Time Scaling Factors.

Jaunt Execution Times

bmark	exec	total (s)	solver (s)	probs	succ	fail	# vars
smol	std	18.18	0.573	14	11	3	4
	nmt	25.51	7.32	16	10	6	25
sconc	std	17.14	0.833	17	14	3	4
	nmt	21.93	8.665	12	10	2	25
mmrxn	std	150.97	6.26	50	50	0	20
	nmt	152.33	23.38	50	50	0	76
gtoggle	std	199.00	4.74	56	56	0	15
	nmt	227.66	45.84	56	54	2	63
repri	std	384.27	12.06	95	87	8	34
	nmt	458.61	95.89	106	95	11	129
bont	std	112.20	4.13	56	49	7	18
	nmt	126.41	21.65	58	49	9	81
epor	std	295.90	9.42	96	81	15	25
	nmt	334.48	71.82	81	76	5	122

Table 6. Jaunt Execution times.