Jonas De Schoenmacker & Frederik Callens

28/11/2019

Abstract

Binnen deze labo-opdracht wordt een Arithmetic and Logic Unit (ALU) ontworpen. Een ALU is een digitaal circuit dat aritmetische operaties (bv. optellen) en logische operaties (bv. bitwise or) kan uitvoeren. De ALU is een fundamenteel onderdeel van elke CPU en microcontroller.

Labo 2

Digitaal ontwerp 1

*Nr. Of pages ????????????*

Table of Contents

[1. Intro 2](#_Toc22474842)

[2. Schema 2](#_Toc22474843)

[3. VHDL 2](#_Toc22474844)

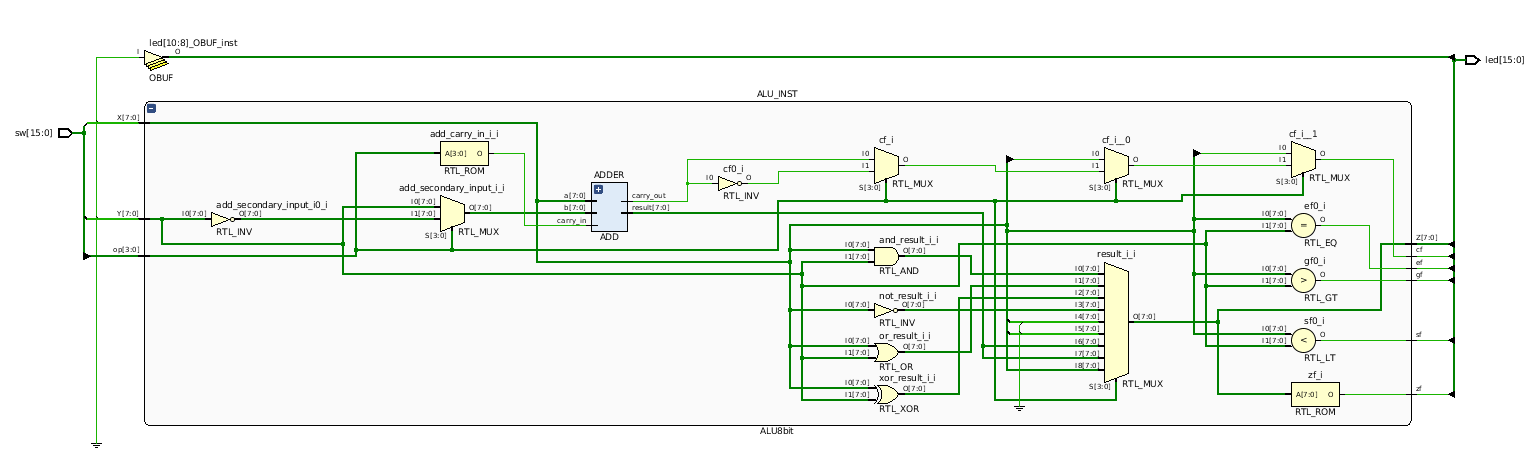
[4. Simulatie output 2](#_Toc22474845)

[5. Besluit 2](#_Toc22474846)

[6. Vragen 2](#_Toc22474847)

# Intro

# Schema



# VHDL

## 3.1 ADD.vhd

----------------------------------------------------------------------------------

-- 𝙸𝚗𝚜𝚝𝚒𝚝𝚞𝚝𝚒𝚘𝚗: 𝙺𝚄 𝙻𝚎𝚞𝚟𝚎𝚗

-- 𝚂𝚝𝚞𝚍𝚎𝚗𝚝𝚜: 𝙹𝚘𝚗𝚊𝚜 𝙳𝚎 𝚂𝚌𝚑𝚘𝚎𝚗𝚖𝚊𝚌𝚔𝚎𝚛 & 𝙵𝚛𝚎𝚍𝚎𝚛𝚒𝚔 𝙲𝚊𝚕𝚕𝚎𝚗𝚜

--

-- 𝙼𝚘𝚍𝚞𝚕𝚎 𝙽𝚊𝚖𝚎: 𝙰𝙳𝙳 - 𝚂𝚝𝚛𝚞𝚌𝚝𝚞𝚛𝚊𝚕

-- 𝙲𝚘𝚞𝚛𝚜𝚎 𝙽𝚊𝚖𝚎: 𝙻𝚊𝚋 𝙳𝚒𝚐𝚒𝚝𝚊𝚕 𝙳𝚎𝚜𝚒𝚐𝚗

--

-- 𝙳𝚎𝚜𝚌𝚛𝚒𝚙𝚝𝚒𝚘𝚗:

-- 𝚗-𝚋𝚒𝚝 𝚛𝚒𝚙𝚙𝚕𝚎 𝚌𝚊𝚛𝚛𝚢 𝚊𝚍𝚍𝚎𝚛

--

----------------------------------------------------------------------------------

𝚕𝚒𝚋𝚛𝚊𝚛𝚢 𝚒𝚎𝚎𝚎;

𝚞𝚜𝚎 𝚒𝚎𝚎𝚎.𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝟷𝟷𝟼𝟺.𝚊𝚕𝚕;

𝚎𝚗𝚝𝚒𝚝𝚢 𝙰𝙳𝙳 𝚒𝚜

𝚐𝚎𝚗𝚎𝚛𝚒𝚌(

𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷 : 𝚗𝚊𝚝𝚞𝚛𝚊𝚕 := 𝟺

);

𝚙𝚘𝚛𝚝(

𝚊 : 𝚒𝚗 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝚟𝚎𝚌𝚝𝚘𝚛((𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷) 𝚍𝚘𝚠𝚗𝚝𝚘 0); -- 𝚒𝚗𝚙𝚞𝚝 𝚟𝚊𝚛 𝟷

𝚋 : 𝚒𝚗 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝚟𝚎𝚌𝚝𝚘𝚛((𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷) 𝚍𝚘𝚠𝚗𝚝𝚘 0); -- 𝚒𝚗𝚙𝚞𝚝 𝚟𝚊𝚛 𝟸

𝚌𝚊𝚛𝚛𝚢\_𝚒𝚗 : 𝚒𝚗 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌; -- 𝚒𝚗𝚙𝚞𝚝 𝚌𝚊𝚛𝚛𝚢

𝚛𝚎𝚜𝚞𝚕𝚝 : 𝚘𝚞𝚝 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝚟𝚎𝚌𝚝𝚘𝚛((𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷) 𝚍𝚘𝚠𝚗𝚝𝚘 0); -- 𝚊𝚕𝚞 𝚘𝚙𝚎𝚛𝚊𝚝𝚒𝚘𝚗 𝚛𝚎𝚜𝚞𝚕𝚝

𝚌𝚊𝚛𝚛𝚢\_𝚘𝚞𝚝 : 𝚘𝚞𝚝 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌 -- 𝚌𝚊𝚛𝚛𝚢

);

𝚎𝚗𝚍 𝚎𝚗𝚝𝚒𝚝𝚢;

𝚊𝚛𝚌𝚑𝚒𝚝𝚎𝚌𝚝𝚞𝚛𝚎 𝙻𝙳𝙳𝟷 𝚘𝚏 𝙰𝙳𝙳 𝚒𝚜

-- 𝚃𝙾𝙳𝙾: 𝚕𝚒𝚜𝚝 𝚘𝚏 𝚜𝚒𝚐𝚗𝚊𝚕𝚜 𝚊𝚗𝚍 𝚌𝚘𝚖𝚙𝚘𝚗𝚎𝚗𝚝𝚜

-- 𝚜𝚒𝚐𝚗𝚊𝚕𝚜

𝚜𝚒𝚐𝚗𝚊𝚕 𝚌 : 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝚟𝚎𝚌𝚝𝚘𝚛(𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷 𝚍𝚘𝚠𝚗𝚝𝚘 0);

-- 𝚌𝚘𝚖𝚙𝚘𝚗𝚎𝚗𝚝𝚜

𝚌𝚘𝚖𝚙𝚘𝚗𝚎𝚗𝚝 𝙵𝙰𝟷𝙱

𝙿𝙾𝚁𝚃(

-- 𝚃𝙾𝙳𝙾: 𝚌𝚘𝚖𝚙𝚕𝚎𝚝𝚎 𝚎𝚗𝚝𝚒𝚝𝚢 𝚍𝚎𝚌𝚕𝚊𝚛𝚊𝚝𝚒𝚘𝚗

𝚊 : 𝚒𝚗 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌; -- 𝚒𝚗𝚙𝚞𝚝 𝚟𝚊𝚛 𝟷

𝚋 : 𝚒𝚗 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌; -- 𝚒𝚗𝚙𝚞𝚝 𝚟𝚊𝚛 𝟸

𝚌𝚊𝚛𝚛𝚢\_𝚒𝚗 : 𝚒𝚗 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌; -- 𝚒𝚗𝚙𝚞𝚝 𝚌𝚊𝚛𝚛𝚢

𝚛𝚎𝚜𝚞𝚕𝚝 : 𝚘𝚞𝚝 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌; -- 𝚊𝚕𝚞 𝚘𝚙𝚎𝚛𝚊𝚝𝚒𝚘𝚗 𝚛𝚎𝚜𝚞𝚕𝚝

𝚌𝚊𝚛𝚛𝚢\_𝚘𝚞𝚝 : 𝚘𝚞𝚝 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌 -- 𝚌𝚊𝚛𝚛𝚢

);

𝚎𝚗𝚍 𝚌𝚘𝚖𝚙𝚘𝚗𝚎𝚗𝚝;

𝚋𝚎𝚐𝚒𝚗

-- 𝚃𝙾𝙳𝙾: 𝚌𝚘𝚖𝚙𝚕𝚎𝚝𝚎 𝚊𝚛𝚌𝚑𝚒𝚝𝚎𝚌𝚝𝚞𝚛𝚎 𝚍𝚎𝚜𝚌𝚛𝚒𝚙𝚝𝚒𝚘𝚗

𝚌(0) <= 𝚌𝚊𝚛𝚛𝚢\_𝚒𝚗;

𝚏𝚘𝚛\_𝚐𝚎𝚗𝚎𝚛𝚊𝚝𝚎: 𝚏𝚘𝚛 𝚒 𝚒𝚗 0 𝚝𝚘 𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷 𝚐𝚎𝚗𝚎𝚛𝚊𝚝𝚎

𝚋𝚎𝚐𝚒𝚗

𝚞𝟷: 𝙵𝙰𝟷𝙱

𝚙𝚘𝚛𝚝 𝚖𝚊𝚙(

𝚊 => 𝚊(𝚒),

𝚋 => 𝚋(𝚒),

𝚌𝚊𝚛𝚛𝚢\_𝚒𝚗 => 𝚌(𝚒),

𝚛𝚎𝚜𝚞𝚕𝚝 => 𝚛𝚎𝚜𝚞𝚕𝚝(𝚒),

𝚌𝚊𝚛𝚛𝚢\_𝚘𝚞𝚝 => 𝚌(𝚒+𝟷)

);

𝚎𝚗𝚍 𝚐𝚎𝚗𝚎𝚛𝚊𝚝𝚎 𝚏𝚘𝚛\_𝚐𝚎𝚗𝚎𝚛𝚊𝚝𝚎;

𝚌𝚊𝚛𝚛𝚢\_𝚘𝚞𝚝 <= 𝚌(𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷);

𝚎𝚗𝚍 𝙻𝙳𝙳𝟷;

## ALU8bit.vhd

----------------------------------------------------------------------------------

-- 𝙸𝚗𝚜𝚝𝚒𝚝𝚞𝚝𝚒𝚘𝚗: 𝙺𝚄 𝙻𝚎𝚞𝚟𝚎𝚗

-- 𝚂𝚝𝚞𝚍𝚎𝚗𝚝𝚜: 𝙵𝚛𝚎𝚍𝚎𝚛𝚒𝚔 𝙲𝚊𝚕𝚕𝚎𝚗𝚜 𝚎𝚗 𝙹𝚘𝚗𝚊𝚜 𝙳𝚎 𝚂𝚌𝚑𝚘𝚎𝚗𝚖𝚊𝚌𝚔𝚎𝚛

--

-- 𝙼𝚘𝚍𝚞𝚕𝚎 𝙽𝚊𝚖𝚎: 𝙰𝙻𝚄𝟾𝚋𝚒𝚝 - 𝙱𝚎𝚑𝚊𝚟𝚒𝚘𝚛𝚊𝚕

-- 𝙲𝚘𝚞𝚛𝚜𝚎 𝙽𝚊𝚖𝚎: 𝙻𝚊𝚋 𝙳𝚒𝚐𝚒𝚝𝚊𝚕 𝙳𝚎𝚜𝚒𝚐𝚗

--

-- 𝙳𝚎𝚜𝚌𝚛𝚒𝚙𝚝𝚒𝚘𝚗:

-- 𝟾-𝚋𝚒𝚝 𝙰𝙻𝚄 𝚝𝚑𝚊𝚝 𝚜𝚞𝚙𝚙𝚘𝚛𝚝𝚜 𝚜𝚎𝚟𝚎𝚛𝚊𝚕 𝚕𝚘𝚐𝚒𝚌 𝚊𝚗𝚍 𝚊𝚛𝚒𝚝𝚑𝚖𝚎𝚝𝚒𝚌 𝚘𝚙𝚎𝚛𝚊𝚝𝚒𝚘𝚗𝚜

--

----------------------------------------------------------------------------------

𝚕𝚒𝚋𝚛𝚊𝚛𝚢 𝙸𝙴𝙴𝙴;

𝚞𝚜𝚎 𝙸𝙴𝙴𝙴.𝚂𝚃𝙳\_𝙻𝙾𝙶𝙸𝙲\_𝟷𝟷𝟼𝟺.𝙰𝙻𝙻;

-- 𝚃𝙾𝙳𝙾: 𝚞𝚜𝚎 𝚙𝚛𝚘𝚌𝚎𝚜𝚜𝚘𝚛\_𝚙𝚔𝚐 𝚏𝚛𝚘𝚖 𝚝𝚑𝚎 𝚠𝚘𝚛𝚔 𝚕𝚒𝚋𝚛𝚊𝚛𝚢

𝚕𝚒𝚋𝚛𝚊𝚛𝚢 𝚠𝚘𝚛𝚔;

𝚞𝚜𝚎 𝚠𝚘𝚛𝚔.𝚙𝚛𝚘𝚌𝚎𝚜𝚜𝚘𝚛\_𝚙𝚔𝚐.𝙰𝙻𝙻;

𝚎𝚗𝚝𝚒𝚝𝚢 𝙰𝙻𝚄𝟾𝚋𝚒𝚝 𝚒𝚜

𝚐𝚎𝚗𝚎𝚛𝚒𝚌(

𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷 : 𝚗𝚊𝚝𝚞𝚛𝚊𝚕 := 𝟾

);

𝚙𝚘𝚛𝚝(

𝚇 : 𝚒𝚗 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝚟𝚎𝚌𝚝𝚘𝚛(𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷 𝚍𝚘𝚠𝚗𝚝𝚘 0);

𝚈 : 𝚒𝚗 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝚟𝚎𝚌𝚝𝚘𝚛(𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷 𝚍𝚘𝚠𝚗𝚝𝚘 0);

𝚉 : 𝚘𝚞𝚝 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝚟𝚎𝚌𝚝𝚘𝚛(𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷 𝚍𝚘𝚠𝚗𝚝𝚘 0);

-- 𝚘𝚙𝚎𝚛𝚊𝚝𝚒𝚘𝚗 𝚜𝚎𝚕𝚎𝚌𝚝

𝚘𝚙 : 𝚒𝚗 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝚟𝚎𝚌𝚝𝚘𝚛(𝟹 𝚍𝚘𝚠𝚗𝚝𝚘 0);

-- 𝚏𝚕𝚊𝚐𝚜

𝚣𝚏 : 𝚘𝚞𝚝 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌;

𝚌𝚏 : 𝚘𝚞𝚝 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌;

𝚎𝚏 : 𝚘𝚞𝚝 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌;

𝚐𝚏 : 𝚘𝚞𝚝 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌;

𝚜𝚏 : 𝚘𝚞𝚝 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌

);

𝚎𝚗𝚍 𝙰𝙻𝚄𝟾𝚋𝚒𝚝;

𝚊𝚛𝚌𝚑𝚒𝚝𝚎𝚌𝚝𝚞𝚛𝚎 𝙱𝚎𝚑𝚊𝚟𝚒𝚘𝚛𝚊𝚕 𝚘𝚏 𝙰𝙻𝚄𝟾𝚋𝚒𝚝 𝚒𝚜

-- 𝚘𝚙𝚎𝚛𝚊𝚝𝚒𝚘𝚗𝚜 𝚍𝚎𝚏𝚒𝚗𝚎𝚍 𝚒𝚗 𝚙𝚛𝚘𝚌𝚎𝚜𝚜𝚘𝚛\_𝚙𝚔𝚐

-- 𝙰𝙻𝚄\_𝙾𝙿\_𝙽𝙾𝚃

-- 𝙰𝙻𝚄\_𝙾𝙿\_𝙰𝙽𝙳

-- 𝙰𝙻𝚄\_𝙾𝙿\_𝙾𝚁

-- 𝙰𝙻𝚄\_𝙾𝙿\_𝚇𝙾𝚁

-- 𝙰𝙻𝚄\_𝙾𝙿\_𝙰𝙳𝙳

-- 𝙰𝙻𝚄\_𝙾𝙿\_𝙲𝙼𝙿

-- 𝙰𝙻𝚄\_𝙾𝙿\_𝚁𝚁

-- 𝙰𝙻𝚄\_𝙾𝙿\_𝚁𝙻

-- 𝙰𝙻𝚄\_𝙾𝙿\_𝚂𝚆𝙰𝙿

-- 𝚘𝚙𝚎𝚛𝚊𝚝𝚒𝚘𝚗 𝚛𝚎𝚜𝚞𝚕𝚝𝚜

𝚜𝚒𝚐𝚗𝚊𝚕 𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 : 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝚟𝚎𝚌𝚝𝚘𝚛(𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷 𝚍𝚘𝚠𝚗𝚝𝚘 0) := (𝚘𝚝𝚑𝚎𝚛𝚜=>'0');

𝚜𝚒𝚐𝚗𝚊𝚕 𝚗𝚘𝚝\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 : 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝚟𝚎𝚌𝚝𝚘𝚛(𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷 𝚍𝚘𝚠𝚗𝚝𝚘 0) := (𝚘𝚝𝚑𝚎𝚛𝚜=>'0');

𝚜𝚒𝚐𝚗𝚊𝚕 𝚊𝚗𝚍\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 : 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝚟𝚎𝚌𝚝𝚘𝚛(𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷 𝚍𝚘𝚠𝚗𝚝𝚘 0) := (𝚘𝚝𝚑𝚎𝚛𝚜=>'0');

𝚜𝚒𝚐𝚗𝚊𝚕 𝚘𝚛\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 : 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝚟𝚎𝚌𝚝𝚘𝚛(𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷 𝚍𝚘𝚠𝚗𝚝𝚘 0) := (𝚘𝚝𝚑𝚎𝚛𝚜=>'0');

𝚜𝚒𝚐𝚗𝚊𝚕 𝚡𝚘𝚛\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 : 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝚟𝚎𝚌𝚝𝚘𝚛(𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷 𝚍𝚘𝚠𝚗𝚝𝚘 0) := (𝚘𝚝𝚑𝚎𝚛𝚜=>'0');

𝚜𝚒𝚐𝚗𝚊𝚕 𝚛𝚛\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 : 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝚟𝚎𝚌𝚝𝚘𝚛(𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷 𝚍𝚘𝚠𝚗𝚝𝚘 0) := (𝚘𝚝𝚑𝚎𝚛𝚜=>'0');

𝚜𝚒𝚐𝚗𝚊𝚕 𝚛𝚕\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 : 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝚟𝚎𝚌𝚝𝚘𝚛(𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷 𝚍𝚘𝚠𝚗𝚝𝚘 0) := (𝚘𝚝𝚑𝚎𝚛𝚜=>'0');

𝚜𝚒𝚐𝚗𝚊𝚕 𝚊𝚍𝚍\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 : 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝚟𝚎𝚌𝚝𝚘𝚛(𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷 𝚍𝚘𝚠𝚗𝚝𝚘 0) := (𝚘𝚝𝚑𝚎𝚛𝚜=>'0');

𝚜𝚒𝚐𝚗𝚊𝚕 𝚜𝚠𝚊𝚙\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 : 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝚟𝚎𝚌𝚝𝚘𝚛(𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷 𝚍𝚘𝚠𝚗𝚝𝚘 0) := (𝚘𝚝𝚑𝚎𝚛𝚜=>'0');

-- 𝚑𝚎𝚕𝚙 𝚜𝚒𝚐𝚗𝚊𝚕𝚜

𝚜𝚒𝚐𝚗𝚊𝚕 𝚊𝚍𝚍\_𝚜𝚎𝚌𝚘𝚗𝚍𝚊𝚛𝚢\_𝚒𝚗𝚙𝚞𝚝\_𝚒 : 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝚟𝚎𝚌𝚝𝚘𝚛(𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷 𝚍𝚘𝚠𝚗𝚝𝚘 0) := (𝚘𝚝𝚑𝚎𝚛𝚜=>'0');

𝚜𝚒𝚐𝚗𝚊𝚕 𝚊𝚍𝚍\_𝚌𝚊𝚛𝚛𝚢\_𝚒𝚗\_𝚒: 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌 := '0';

𝚜𝚒𝚐𝚗𝚊𝚕 𝚊𝚍𝚍\_𝚌𝚊𝚛𝚛𝚢\_𝚒 : 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌 := '0';

𝚜𝚒𝚐𝚗𝚊𝚕 𝚛𝚛\_𝚌𝚊𝚛𝚛𝚢: 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌 := '0';

𝚜𝚒𝚐𝚗𝚊𝚕 𝚛𝚕\_𝚌𝚊𝚛𝚛𝚢: 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌 := '0';

𝚜𝚒𝚐𝚗𝚊𝚕 𝚣𝚎𝚛𝚘𝚜: 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝚟𝚎𝚌𝚝𝚘𝚛(𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷 𝚍𝚘𝚠𝚗𝚝𝚘 0) := (𝚘𝚝𝚑𝚎𝚛𝚜=>'0');

-- 𝚠𝚎 𝚞𝚜𝚎 𝚊 𝚜𝚎𝚙𝚊𝚛𝚊𝚝𝚎 𝚖𝚘𝚍𝚞𝚕𝚎 𝚏𝚘𝚛 𝚝𝚑𝚎 𝚊𝚍𝚍𝚒𝚝𝚒𝚘𝚗/𝚜𝚞𝚋𝚝𝚛𝚊𝚌𝚝𝚒𝚘𝚗

𝚌𝚘𝚖𝚙𝚘𝚗𝚎𝚗𝚝 𝙰𝙳𝙳 𝚒𝚜

𝚐𝚎𝚗𝚎𝚛𝚒𝚌(

𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷 : 𝚗𝚊𝚝𝚞𝚛𝚊𝚕 := 𝟺

);

𝚙𝚘𝚛𝚝(

𝚊 : 𝚒𝚗 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝚟𝚎𝚌𝚝𝚘𝚛((𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷) 𝚍𝚘𝚠𝚗𝚝𝚘 0); -- 𝚒𝚗𝚙𝚞𝚝 𝚟𝚊𝚛 𝟷

𝚋 : 𝚒𝚗 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝚟𝚎𝚌𝚝𝚘𝚛((𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷) 𝚍𝚘𝚠𝚗𝚝𝚘 0); -- 𝚒𝚗𝚙𝚞𝚝 𝚟𝚊𝚛 𝟸

𝚌𝚊𝚛𝚛𝚢\_𝚒𝚗 : 𝚒𝚗 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌; -- 𝚒𝚗𝚙𝚞𝚝 𝚌𝚊𝚛𝚛𝚢

𝚛𝚎𝚜𝚞𝚕𝚝 : 𝚘𝚞𝚝 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝚟𝚎𝚌𝚝𝚘𝚛((𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷) 𝚍𝚘𝚠𝚗𝚝𝚘 0); -- 𝚊𝚕𝚞 𝚘𝚙𝚎𝚛𝚊𝚝𝚒𝚘𝚗 𝚛𝚎𝚜𝚞𝚕𝚝

𝚌𝚊𝚛𝚛𝚢\_𝚘𝚞𝚝 : 𝚘𝚞𝚝 𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌 -- 𝚌𝚊𝚛𝚛𝚢

);

𝚎𝚗𝚍 𝚌𝚘𝚖𝚙𝚘𝚗𝚎𝚗𝚝;

𝚋𝚎𝚐𝚒𝚗

-- 𝚃𝙾𝙳𝙾: 𝚌𝚘𝚖𝚙𝚕𝚎𝚝𝚎 𝚝𝚑𝚎 𝚏𝚘𝚕𝚕𝚘𝚠𝚒𝚗𝚐 𝚕𝚒𝚗𝚎𝚜 𝚝𝚘 𝚙𝚎𝚛𝚏𝚘𝚛𝚖 𝚕𝚘𝚐𝚒𝚌𝚊𝚕 𝚘𝚙𝚎𝚛𝚊𝚝𝚒𝚘𝚗𝚜

-- 𝚒𝚖𝚙𝚕𝚎𝚖𝚎𝚗𝚝𝚊𝚝𝚒𝚘𝚗 𝚘𝚏 𝚜𝚘𝚖𝚎 𝚘𝚙𝚎𝚛𝚊𝚝𝚒𝚘𝚗𝚜

-- 𝚊𝚗𝚍

𝚊𝚗𝚍\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 <= 𝚇 𝚊𝚗𝚍 𝚈;

-- 𝚘𝚛

𝚘𝚛\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 <= 𝚇 𝚘𝚛 𝚈;

-- 𝚡𝚘𝚛

𝚡𝚘𝚛\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 <= 𝚇 𝚡𝚘𝚛 𝚈;

-- 𝚗𝚘𝚝

𝚗𝚘𝚝\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 <= 𝚗𝚘𝚝(𝚇) ;

-- 𝚛𝚛

𝚛𝚛\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 <= '0' & 𝚇(𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷 𝚍𝚘𝚠𝚗𝚝𝚘 𝟷);

𝚛𝚛\_𝚌𝚊𝚛𝚛𝚢 <= 𝚇(0);

-- 𝚛𝚕

𝚛𝚕\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 <= 𝚇(𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟸 𝚍𝚘𝚠𝚗𝚝𝚘 0) & '0';

𝚛𝚕\_𝚌𝚊𝚛𝚛𝚢 <= 𝚇(𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷);

-- 𝚜𝚠𝚊𝚙

𝚜𝚠𝚊𝚙\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 <= 𝚇((𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷/𝟸)-𝟷 𝚍𝚘𝚠𝚗𝚝𝚘 0) & 𝚇(𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷-𝟷 𝚍𝚘𝚠𝚗𝚝𝚘 𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷/𝟸);

-- 𝚃𝙾𝙳𝙾: 𝚑𝚊𝚟𝚎 𝚊 𝚕𝚘𝚘𝚔 𝚊𝚝 𝚑𝚘𝚠 𝚝𝚑𝚒𝚜 𝚖𝚘𝚍𝚞𝚕𝚎 𝚒𝚜 𝚒𝚗𝚜𝚝𝚊𝚗𝚝𝚒𝚊𝚝𝚎𝚍

-- 𝚁𝚒𝚙𝚙𝚕𝚎 𝚌𝚊𝚛𝚛𝚢 𝚊𝚍𝚍𝚎𝚛 𝚒𝚗𝚜𝚝𝚊𝚗𝚝𝚒𝚊𝚝𝚒𝚘𝚗

𝙰𝙳𝙳𝙴𝚁 : 𝙰𝙳𝙳

𝚐𝚎𝚗𝚎𝚛𝚒𝚌 𝚖𝚊𝚙(

𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷 => 𝙲\_𝙳𝙰𝚃𝙰\_𝚆𝙸𝙳𝚃𝙷 -- 𝚝𝚑𝚒𝚜 𝚠𝚒𝚕𝚕 𝚌𝚑𝚊𝚗𝚐𝚎 𝚝𝚑𝚎 𝚍𝚎𝚏𝚊𝚞𝚕𝚝 𝚠𝚒𝚍𝚝𝚑 𝚘𝚏 𝚝𝚑𝚎 𝚊𝚍𝚍𝚎𝚛 𝚝𝚘 𝚝𝚑𝚎 𝚠𝚒𝚍𝚝𝚑 𝚜𝚙𝚎𝚌𝚒𝚏𝚒𝚎𝚍 𝚑𝚎𝚛𝚎

)

𝚙𝚘𝚛𝚝 𝚖𝚊𝚙(

𝚊 => 𝚇,

𝚋 => 𝚊𝚍𝚍\_𝚜𝚎𝚌𝚘𝚗𝚍𝚊𝚛𝚢\_𝚒𝚗𝚙𝚞𝚝\_𝚒,

𝚌𝚊𝚛𝚛𝚢\_𝚒𝚗 => 𝚊𝚍𝚍\_𝚌𝚊𝚛𝚛𝚢\_𝚒𝚗\_𝚒,

𝚛𝚎𝚜𝚞𝚕𝚝 => 𝚊𝚍𝚍\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒,

𝚌𝚊𝚛𝚛𝚢\_𝚘𝚞𝚝 => 𝚊𝚍𝚍\_𝚌𝚊𝚛𝚛𝚢\_𝚒

);

-- 𝚃𝙾𝙳𝙾: 𝚌𝚑𝚊𝚗𝚐𝚎 𝚝𝚑𝚎 𝚊𝚍𝚍𝚎𝚛'𝚜 𝚜𝚎𝚌𝚘𝚗𝚍𝚊𝚛𝚢 𝚒𝚗𝚙𝚞𝚝 𝚊𝚗𝚍 𝚌𝚊𝚛𝚛𝚢 𝚒𝚗, 𝚋𝚊𝚜𝚎𝚍 𝚘𝚗 𝚝𝚑𝚎 𝚘𝚙𝚎𝚛𝚊𝚝𝚒𝚘𝚗 (𝚊𝚍𝚍𝚒𝚝𝚒𝚘𝚗/𝚜𝚞𝚋𝚝𝚛𝚊𝚌𝚝𝚒𝚘𝚗)

-- 𝚊𝚍𝚍𝚒𝚝𝚒𝚘𝚗 𝚊𝚗𝚍 𝚜𝚞𝚋𝚝𝚛𝚊𝚌𝚝𝚒𝚘𝚗

𝚊𝚍𝚍\_𝚜𝚎𝚌𝚘𝚗𝚍𝚊𝚛𝚢\_𝚒𝚗𝚙𝚞𝚝\_𝚒 <= 𝚈 𝚠𝚑𝚎𝚗 (𝚘𝚙=𝙰𝙻𝚄\_𝙾𝙿\_𝙰𝙳𝙳) 𝚎𝚕𝚜𝚎 𝚗𝚘𝚝(𝚈);

𝚊𝚍𝚍\_𝚌𝚊𝚛𝚛𝚢\_𝚒𝚗\_𝚒 <= '0' 𝚠𝚑𝚎𝚗 (𝚘𝚙=𝙰𝙻𝚄\_𝙾𝙿\_𝙰𝙳𝙳) 𝚎𝚕𝚜𝚎 '𝟷';

-- 𝚃𝙾𝙳𝙾: 𝚜𝚎𝚝 '𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒' 𝚝𝚘 𝚊 𝚜𝚙𝚎𝚌𝚒𝚏𝚒𝚌 𝚘𝚙𝚎𝚛𝚊𝚝𝚒𝚘𝚗 𝚛𝚎𝚜𝚞𝚕𝚝 𝚋𝚊𝚜𝚎𝚍 𝚘𝚗 𝚝𝚑𝚎 𝚜𝚎𝚕𝚎𝚌𝚝𝚎𝚍 𝚘𝚙𝚎𝚛𝚊𝚝𝚒𝚘𝚗 '𝚘𝚙'

-- 𝚛𝚎𝚜𝚞𝚕𝚝 𝚖𝚞𝚡:

𝚠𝚒𝚝𝚑 𝚘𝚙 𝚜𝚎𝚕𝚎𝚌𝚝

𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 <= 𝚊𝚗𝚍\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 𝚠𝚑𝚎𝚗 𝙰𝙻𝚄\_𝙾𝙿\_𝙰𝙽𝙳,

𝚘𝚛\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 𝚠𝚑𝚎𝚗 𝙰𝙻𝚄\_𝙾𝙿\_𝙾𝚁,

𝚡𝚘𝚛\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 𝚠𝚑𝚎𝚗 𝙰𝙻𝚄\_𝙾𝙿\_𝚇𝙾𝚁,

𝚗𝚘𝚝\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 𝚠𝚑𝚎𝚗 𝙰𝙻𝚄\_𝙾𝙿\_𝙽𝙾𝚃,

𝚛𝚛\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 𝚠𝚑𝚎𝚗 𝙰𝙻𝚄\_𝙾𝙿\_𝚁𝚁,

𝚛𝚕\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 𝚠𝚑𝚎𝚗 𝙰𝙻𝚄\_𝙾𝙿\_𝚁𝙻,

𝚊𝚍𝚍\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 𝚠𝚑𝚎𝚗 𝙰𝙻𝚄\_𝙾𝙿\_𝙰𝙳𝙳,

𝚊𝚍𝚍\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 𝚠𝚑𝚎𝚗 𝙰𝙻𝚄\_𝙾𝙿\_𝚂𝚄𝙱,

𝚜𝚠𝚊𝚙\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 𝚠𝚑𝚎𝚗 𝚘𝚝𝚑𝚎𝚛𝚜;

-- 𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 <= 𝚊𝚗𝚍\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 𝚠𝚑𝚎𝚗 (𝚘𝚙=𝙰𝙻𝚄\_𝙾𝙿\_𝙰𝙽𝙳) 𝚎𝚕𝚜𝚎

-- 𝚘𝚛\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 𝚠𝚑𝚎𝚗 (𝚘𝚙=𝙰𝙻𝚄\_𝙾𝙿\_𝙾𝚁) 𝚎𝚕𝚜𝚎

-- 𝚡𝚘𝚛\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 𝚠𝚑𝚎𝚗 (𝚘𝚙=𝙰𝙻𝚄\_𝙾𝙿\_𝚇𝙾𝚁) 𝚎𝚕𝚜𝚎

-- 𝚗𝚘𝚝\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 𝚠𝚑𝚎𝚗 (𝚘𝚙=𝙰𝙻𝚄\_𝙾𝙿\_𝙽𝙾𝚃) 𝚎𝚕𝚜𝚎

-- 𝚛𝚛\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 𝚠𝚑𝚎𝚗 (𝚘𝚙=𝙰𝙻𝚄\_𝙾𝙿\_𝚁𝚁) 𝚎𝚕𝚜𝚎

-- 𝚛𝚕\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 𝚠𝚑𝚎𝚗 (𝚘𝚙=𝙰𝙻𝚄\_𝙾𝙿\_𝚁𝙻) 𝚎𝚕𝚜𝚎

-- 𝚊𝚍𝚍\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 𝚠𝚑𝚎𝚗 (𝚘𝚙=𝙰𝙻𝚄\_𝙾𝙿\_𝙰𝙳𝙳) 𝚎𝚕𝚜𝚎

-- 𝚊𝚍𝚍\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 𝚠𝚑𝚎𝚗 (𝚘𝚙=𝙰𝙻𝚄\_𝙾𝙿\_𝚂𝚄𝙱) 𝚎𝚕𝚜𝚎

-- 𝚜𝚠𝚊𝚙\_𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒;

𝚉 <= 𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒;

-- 𝚊𝚍𝚍\_𝚌𝚊𝚛𝚛𝚢\_𝚒 <= 𝚗𝚘𝚝(𝚊𝚍𝚍\_𝚌𝚊𝚛𝚛𝚢\_𝚒) 𝚠𝚑𝚎𝚗 (𝚘𝚙=𝙰𝙻𝚄\_𝙾𝙿\_𝚂𝚄𝙱) 𝚎𝚕𝚜𝚎

-- 𝚊𝚍𝚍\_𝚌𝚊𝚛𝚛𝚢\_𝚒;

-- 𝚃𝙾𝙳𝙾: 𝚌𝚘𝚗𝚝𝚛𝚘𝚕 𝚝𝚑𝚎 𝚏𝚕𝚊𝚐𝚜

-- 𝚌𝚊𝚛𝚛𝚢 𝚏𝚕𝚊𝚐: 𝟷 𝚌𝚊𝚛𝚛𝚢 𝚏𝚕𝚊𝚐 𝚏𝚘𝚛 𝚂𝚄𝙱, 𝙰𝙳𝙳, 𝚁𝚁 𝚊𝚗𝚍 𝚁𝙻 (𝚋𝚊𝚜𝚎𝚍 𝚘𝚗 𝚘𝚙)

-- 𝚍𝚘𝚗'𝚝 𝚏𝚘𝚛𝚐𝚎𝚝 𝚝𝚑𝚊𝚝 𝚛𝚘𝚝𝚊𝚝𝚎 𝚕𝚎𝚏𝚝/𝚛𝚒𝚐𝚑𝚝 𝚌𝚊𝚗 𝚊𝚕𝚜𝚘 𝚙𝚛𝚘𝚍𝚞𝚌𝚎 𝚊 𝚌𝚊𝚛𝚛𝚢

-- 𝚢𝚘𝚞 𝚖𝚒𝚐𝚑𝚝 𝚗𝚎𝚎𝚍 𝚜𝚘𝚖𝚎 𝚎𝚡𝚝𝚛𝚊 𝚜𝚒𝚐𝚗𝚊𝚕𝚜

𝚌𝚏 <= 𝚛𝚛\_𝚌𝚊𝚛𝚛𝚢 𝚠𝚑𝚎𝚗 (𝚘𝚙=𝙰𝙻𝚄\_𝙾𝙿\_𝚁𝚁) 𝚎𝚕𝚜𝚎

𝚛𝚕\_𝚌𝚊𝚛𝚛𝚢 𝚠𝚑𝚎𝚗 (𝚘𝚙=𝙰𝙻𝚄\_𝙾𝙿\_𝚁𝙻) 𝚎𝚕𝚜𝚎

𝚊𝚍𝚍\_𝚌𝚊𝚛𝚛𝚢\_𝚒 𝚠𝚑𝚎𝚗 (𝚘𝚙=𝙰𝙻𝚄\_𝙾𝙿\_𝙰𝙳𝙳) 𝚎𝚕𝚜𝚎

𝚗𝚘𝚝(𝚊𝚍𝚍\_𝚌𝚊𝚛𝚛𝚢\_𝚒);

-- 𝚣𝚎𝚛𝚘 𝚏𝚕𝚊𝚐

𝚣𝚏 <= '𝟷' 𝚠𝚑𝚎𝚗 𝚛𝚎𝚜𝚞𝚕𝚝\_𝚒 = 𝚣𝚎𝚛𝚘𝚜 𝚎𝚕𝚜𝚎 '0';

-- 𝚎𝚚𝚞𝚊𝚕, 𝚜𝚖𝚊𝚕𝚕𝚎𝚛, 𝚐𝚛𝚎𝚊𝚝𝚎𝚛 𝚏𝚕𝚊𝚐

𝚎𝚏 <= '𝟷' 𝚠𝚑𝚎𝚗 𝚇=𝚈 𝚎𝚕𝚜𝚎 '0';

𝚐𝚏 <= '𝟷' 𝚠𝚑𝚎𝚗 𝚇>𝚈 𝚎𝚕𝚜𝚎 '0';

𝚜𝚏 <= '𝟷' 𝚠𝚑𝚎𝚗 𝚇<𝚈 𝚎𝚕𝚜𝚎 '0';

𝚎𝚗𝚍 𝙱𝚎𝚑𝚊𝚟𝚒𝚘𝚛𝚊𝚕;

# Simulatie output

## Simulation output wave signal



## Simulation output tcl console ADDER

𝚜𝚘𝚞𝚛𝚌𝚎 /𝚑𝚘𝚖𝚎/𝚜𝚝𝚞𝚍𝚎𝚗𝚝/𝙻𝚊𝚋𝚘𝟸/𝙻𝚊𝚋𝚘𝟸.𝚜𝚒𝚖/𝚜𝚒𝚖\_𝟷/𝚋𝚎𝚑𝚊𝚟/𝚡𝚜𝚒𝚖/𝚡𝚜𝚒𝚖.𝚍𝚒𝚛/𝙰𝙳𝙳\_𝚝𝚋\_𝚋𝚎𝚑𝚊𝚟/𝚠𝚎𝚋𝚝𝚊𝚕𝚔/𝚡𝚜𝚒𝚖\_𝚠𝚎𝚋𝚝𝚊𝚕𝚔.𝚝𝚌𝚕 -𝚗𝚘𝚝𝚛𝚊𝚌𝚎

𝙸𝙽𝙵𝙾: [𝙲𝚘𝚖𝚖𝚘𝚗 𝟷𝟽-𝟷𝟾𝟼] '/𝚑𝚘𝚖𝚎/𝚜𝚝𝚞𝚍𝚎𝚗𝚝/𝙻𝚊𝚋𝚘𝟸/𝙻𝚊𝚋𝚘𝟸.𝚜𝚒𝚖/𝚜𝚒𝚖\_𝟷/𝚋𝚎𝚑𝚊𝚟/𝚡𝚜𝚒𝚖/𝚡𝚜𝚒𝚖.𝚍𝚒𝚛/𝙰𝙳𝙳\_𝚝𝚋\_𝚋𝚎𝚑𝚊𝚟/𝚠𝚎𝚋𝚝𝚊𝚕𝚔/𝚞𝚜𝚊𝚐𝚎\_𝚜𝚝𝚊𝚝𝚒𝚜𝚝𝚒𝚌𝚜\_𝚎𝚡𝚝\_𝚡𝚜𝚒𝚖.𝚡𝚖𝚕' 𝚑𝚊𝚜 𝚋𝚎𝚎𝚗 𝚜𝚞𝚌𝚌𝚎𝚜𝚜𝚏𝚞𝚕𝚕𝚢 𝚜𝚎𝚗𝚝 𝚝𝚘 𝚇𝚒𝚕𝚒𝚗𝚡 𝚘𝚗 𝚃𝚞𝚎 𝙾𝚌𝚝 𝟷𝟻 𝟷𝟺:𝟻𝟼:𝟸𝟹 𝟸0𝟷𝟿. 𝙵𝚘𝚛 𝚊𝚍𝚍𝚒𝚝𝚒𝚘𝚗𝚊𝚕 𝚍𝚎𝚝𝚊𝚒𝚕𝚜 𝚊𝚋𝚘𝚞𝚝 𝚝𝚑𝚒𝚜 𝚏𝚒𝚕𝚎, 𝚙𝚕𝚎𝚊𝚜𝚎 𝚛𝚎𝚏𝚎𝚛 𝚝𝚘 𝚝𝚑𝚎 𝚆𝚎𝚋𝚃𝚊𝚕𝚔 𝚑𝚎𝚕𝚙 𝚏𝚒𝚕𝚎 𝚊𝚝 /𝚘𝚙𝚝/𝚇𝚒𝚕𝚒𝚗𝚡/𝚅𝚒𝚟𝚊𝚍𝚘/𝟸0𝟷𝟽.𝟺/𝚍𝚘𝚌/𝚠𝚎𝚋𝚝𝚊𝚕𝚔\_𝚒𝚗𝚝𝚛𝚘𝚍𝚞𝚌𝚝𝚒𝚘𝚗.𝚑𝚝𝚖𝚕.

𝙸𝙽𝙵𝙾: [𝙲𝚘𝚖𝚖𝚘𝚗 𝟷𝟽-𝟸0𝟼] 𝙴𝚡𝚒𝚝𝚒𝚗𝚐 𝚆𝚎𝚋𝚝𝚊𝚕𝚔 𝚊𝚝 𝚃𝚞𝚎 𝙾𝚌𝚝 𝟷𝟻 𝟷𝟺:𝟻𝟼:𝟸𝟹 𝟸0𝟷𝟿...

𝚛𝚞𝚗\_𝚙𝚛𝚘𝚐𝚛𝚊𝚖: 𝚃𝚒𝚖𝚎 (𝚜): 𝚌𝚙𝚞 = 00:00:0𝟸 ; 𝚎𝚕𝚊𝚙𝚜𝚎𝚍 = 00:00:0𝟾 . 𝙼𝚎𝚖𝚘𝚛𝚢 (𝙼𝙱): 𝚙𝚎𝚊𝚔 = 𝟼𝟻𝟾𝟽.𝟿𝟿𝟸 ; 𝚐𝚊𝚒𝚗 = 0.000 ; 𝚏𝚛𝚎𝚎 𝚙𝚑𝚢𝚜𝚒𝚌𝚊𝚕 = 𝟸𝟼𝟹𝟾 ; 𝚏𝚛𝚎𝚎 𝚟𝚒𝚛𝚝𝚞𝚊𝚕 = 𝟷𝟹𝟹𝟾𝟽

𝙸𝙽𝙵𝙾: [𝚄𝚂𝙵-𝚇𝚂𝚒𝚖-𝟼𝟿] '𝚎𝚕𝚊𝚋𝚘𝚛𝚊𝚝𝚎' 𝚜𝚝𝚎𝚙 𝚏𝚒𝚗𝚒𝚜𝚑𝚎𝚍 𝚒𝚗 '𝟾' 𝚜𝚎𝚌𝚘𝚗𝚍𝚜

𝙸𝙽𝙵𝙾: [𝚄𝚂𝙵-𝚇𝚂𝚒𝚖-𝟺] 𝚇𝚂𝚒𝚖::𝚂𝚒𝚖𝚞𝚕𝚊𝚝𝚎 𝚍𝚎𝚜𝚒𝚐𝚗

𝙸𝙽𝙵𝙾: [𝚄𝚂𝙵-𝚇𝚂𝚒𝚖-𝟼𝟷] 𝙴𝚡𝚎𝚌𝚞𝚝𝚒𝚗𝚐 '𝚂𝙸𝙼𝚄𝙻𝙰𝚃𝙴' 𝚜𝚝𝚎𝚙 𝚒𝚗 '/𝚑𝚘𝚖𝚎/𝚜𝚝𝚞𝚍𝚎𝚗𝚝/𝙻𝚊𝚋𝚘𝟸/𝙻𝚊𝚋𝚘𝟸.𝚜𝚒𝚖/𝚜𝚒𝚖\_𝟷/𝚋𝚎𝚑𝚊𝚟/𝚡𝚜𝚒𝚖'

𝙸𝙽𝙵𝙾: [𝚄𝚂𝙵-𝚇𝚂𝚒𝚖-𝟿𝟾] \*\*\* 𝚁𝚞𝚗𝚗𝚒𝚗𝚐 𝚡𝚜𝚒𝚖

𝚠𝚒𝚝𝚑 𝚊𝚛𝚐𝚜 "𝙰𝙳𝙳\_𝚝𝚋\_𝚋𝚎𝚑𝚊𝚟 -𝚔𝚎𝚢 {𝙱𝚎𝚑𝚊𝚟𝚒𝚘𝚛𝚊𝚕:𝚜𝚒𝚖\_𝟷:𝙵𝚞𝚗𝚌𝚝𝚒𝚘𝚗𝚊𝚕:𝙰𝙳𝙳\_𝚝𝚋} -𝚝𝚌𝚕𝚋𝚊𝚝𝚌𝚑 {𝙰𝙳𝙳\_𝚝𝚋.𝚝𝚌𝚕} -𝚕𝚘𝚐 {𝚜𝚒𝚖𝚞𝚕𝚊𝚝𝚎.𝚕𝚘𝚐}"

𝙸𝙽𝙵𝙾: [𝚄𝚂𝙵-𝚇𝚂𝚒𝚖-𝟾] 𝙻𝚘𝚊𝚍𝚒𝚗𝚐 𝚜𝚒𝚖𝚞𝚕𝚊𝚝𝚘𝚛 𝚏𝚎𝚊𝚝𝚞𝚛𝚎

𝚅𝚒𝚟𝚊𝚍𝚘 𝚂𝚒𝚖𝚞𝚕𝚊𝚝𝚘𝚛 𝟸0𝟷𝟽.𝟺

𝚃𝚒𝚖𝚎 𝚛𝚎𝚜𝚘𝚕𝚞𝚝𝚒𝚘𝚗 𝚒𝚜 𝟷 𝚙𝚜

𝚜𝚘𝚞𝚛𝚌𝚎 𝙰𝙳𝙳\_𝚝𝚋.𝚝𝚌𝚕

# 𝚜𝚎𝚝 𝚌𝚞𝚛𝚛\_𝚠𝚊𝚟𝚎 [𝚌𝚞𝚛𝚛𝚎𝚗𝚝\_𝚠𝚊𝚟𝚎\_𝚌𝚘𝚗𝚏𝚒𝚐]

# 𝚒𝚏 { [𝚜𝚝𝚛𝚒𝚗𝚐 𝚕𝚎𝚗𝚐𝚝𝚑 $𝚌𝚞𝚛𝚛\_𝚠𝚊𝚟𝚎] == 0 } {

# 𝚒𝚏 { [𝚕𝚕𝚎𝚗𝚐𝚝𝚑 [𝚐𝚎𝚝\_𝚘𝚋𝚓𝚎𝚌𝚝𝚜]] > 0} {

# 𝚊𝚍𝚍\_𝚠𝚊𝚟𝚎 /

# 𝚜𝚎𝚝\_𝚙𝚛𝚘𝚙𝚎𝚛𝚝𝚢 𝚗𝚎𝚎𝚍𝚜\_𝚜𝚊𝚟𝚎 𝚏𝚊𝚕𝚜𝚎 [𝚌𝚞𝚛𝚛𝚎𝚗𝚝\_𝚠𝚊𝚟𝚎\_𝚌𝚘𝚗𝚏𝚒𝚐]

# } 𝚎𝚕𝚜𝚎 {

# 𝚜𝚎𝚗𝚍\_𝚖𝚜𝚐\_𝚒𝚍 𝙰𝚍𝚍\_𝚆𝚊𝚟𝚎-𝟷 𝚆𝙰𝚁𝙽𝙸𝙽𝙶 "𝙽𝚘 𝚝𝚘𝚙 𝚕𝚎𝚟𝚎𝚕 𝚜𝚒𝚐𝚗𝚊𝚕𝚜 𝚏𝚘𝚞𝚗𝚍. 𝚂𝚒𝚖𝚞𝚕𝚊𝚝𝚘𝚛 𝚠𝚒𝚕𝚕 𝚜𝚝𝚊𝚛𝚝 𝚠𝚒𝚝𝚑𝚘𝚞𝚝 𝚊 𝚠𝚊𝚟𝚎 𝚠𝚒𝚗𝚍𝚘𝚠. 𝙸𝚏 𝚢𝚘𝚞 𝚠𝚊𝚗𝚝 𝚝𝚘 𝚘𝚙𝚎𝚗 𝚊 𝚠𝚊𝚟𝚎 𝚠𝚒𝚗𝚍𝚘𝚠 𝚐𝚘 𝚝𝚘 '𝙵𝚒𝚕𝚎->𝙽𝚎𝚠 𝚆𝚊𝚟𝚎𝚏𝚘𝚛𝚖 𝙲𝚘𝚗𝚏𝚒𝚐𝚞𝚛𝚊𝚝𝚒𝚘𝚗' 𝚘𝚛 𝚝𝚢𝚙𝚎 '𝚌𝚛𝚎𝚊𝚝𝚎\_𝚠𝚊𝚟𝚎\_𝚌𝚘𝚗𝚏𝚒𝚐' 𝚒𝚗 𝚝𝚑𝚎 𝚃𝙲𝙻 𝚌𝚘𝚗𝚜𝚘𝚕𝚎."

# }

# }

# 𝚛𝚞𝚗 𝟷000𝚗𝚜

𝚂𝚄𝙲𝙲𝙴𝚂𝚂: 𝙰𝙳𝙳 𝚠𝚒𝚝𝚑𝚘𝚞𝚝 𝚌𝚊𝚛𝚛𝚢.

𝚂𝚄𝙲𝙲𝙴𝚂𝚂: 𝙰𝙳𝙳+𝚌𝚊𝚛𝚛𝚢\_𝚒𝚗 𝚠𝚒𝚝𝚑𝚘𝚞𝚝 𝚌𝚊𝚛𝚛𝚢.

𝚂𝚄𝙲𝙲𝙴𝚂𝚂: 𝙰𝙳𝙳 𝚠𝚒𝚝𝚑 𝚌𝚊𝚛𝚛𝚢.

𝚂𝚄𝙲𝙲𝙴𝚂𝚂: 𝙰𝙳𝙳+𝚌𝚊𝚛𝚛𝚢\_𝚒𝚗 𝚠𝚒𝚝𝚑 𝚌𝚊𝚛𝚛𝚢.

𝙽𝚘𝚝𝚎: 𝚂𝙸𝙼𝚄𝙻𝙰𝚃𝙸𝙾𝙽 𝙴𝙽𝙳𝙴𝙳

𝚃𝚒𝚖𝚎: 𝟼𝟼 𝚗𝚜 𝙸𝚝𝚎𝚛𝚊𝚝𝚒𝚘𝚗: 0 𝙿𝚛𝚘𝚌𝚎𝚜𝚜: /𝙰𝙳𝙳\_𝚝𝚋/𝚂𝚃𝙸𝙼\_𝙿𝚁𝙾𝙲 𝙵𝚒𝚕𝚎: /𝚑𝚘𝚖𝚎/𝚜𝚝𝚞𝚍𝚎𝚗𝚝/𝙻𝚊𝚋𝚘𝟸/𝙻𝚊𝚋𝚘𝟸.𝚜𝚛𝚌𝚜/𝚜𝚒𝚖\_𝟷/𝚒𝚖𝚙𝚘𝚛𝚝𝚜/𝚜𝚘𝚞𝚛𝚌𝚎𝚜/𝙰𝙳𝙳\_𝚝𝚋.𝚟𝚑𝚍

𝙸𝙽𝙵𝙾: [𝚄𝚂𝙵-𝚇𝚂𝚒𝚖-𝟿𝟼] 𝚇𝚂𝚒𝚖 𝚌𝚘𝚖𝚙𝚕𝚎𝚝𝚎𝚍. 𝙳𝚎𝚜𝚒𝚐𝚗 𝚜𝚗𝚊𝚙𝚜𝚑𝚘𝚝 '𝙰𝙳𝙳\_𝚝𝚋\_𝚋𝚎𝚑𝚊𝚟' 𝚕𝚘𝚊𝚍𝚎𝚍.

𝙸𝙽𝙵𝙾: [𝚄𝚂𝙵-𝚇𝚂𝚒𝚖-𝟿𝟽] 𝚇𝚂𝚒𝚖 𝚜𝚒𝚖𝚞𝚕𝚊𝚝𝚒𝚘𝚗 𝚛𝚊𝚗 𝚏𝚘𝚛 𝟷000𝚗𝚜

𝚕𝚊𝚞𝚗𝚌𝚑\_𝚜𝚒𝚖𝚞𝚕𝚊𝚝𝚒𝚘𝚗: 𝚃𝚒𝚖𝚎 (𝚜): 𝚌𝚙𝚞 = 00:00:0𝟽 ; 𝚎𝚕𝚊𝚙𝚜𝚎𝚍 = 00:00:𝟷𝟸 . 𝙼𝚎𝚖𝚘𝚛𝚢 (𝙼𝙱): 𝚙𝚎𝚊𝚔 = 𝟼𝟼𝟼𝟽.𝟽𝟺𝟸 ; 𝚐𝚊𝚒𝚗 = 𝟽𝟿.𝟽𝟻0 ; 𝚏𝚛𝚎𝚎 𝚙𝚑𝚢𝚜𝚒𝚌𝚊𝚕 = 𝟸𝟻𝟿𝟷 ; 𝚏𝚛𝚎𝚎 𝚟𝚒𝚛𝚝𝚞𝚊𝚕 = 𝟷𝟹𝟹𝟻𝟷

## Simulation output tcl console ALU

𝚛𝚎𝚕𝚊𝚞𝚗𝚌𝚑\_𝚜𝚒𝚖: 𝚃𝚒𝚖𝚎 (𝚜): 𝚌𝚙𝚞 = 00:00:0𝟹 ; 𝚎𝚕𝚊𝚙𝚜𝚎𝚍 = 00:00:0𝟻 . 𝙼𝚎𝚖𝚘𝚛𝚢 (𝙼𝙱): 𝚙𝚎𝚊𝚔 = 𝟼𝟽𝟾𝟻.𝟾𝟺𝟺 ; 𝚐𝚊𝚒𝚗 = 0.000 ; 𝚏𝚛𝚎𝚎 𝚙𝚑𝚢𝚜𝚒𝚌𝚊𝚕 = 𝟸𝟻𝟼𝟹 ; 𝚏𝚛𝚎𝚎 𝚟𝚒𝚛𝚝𝚞𝚊𝚕 = 𝟷𝟹𝟸𝟽𝟼

𝚛𝚎𝚕𝚊𝚞𝚗𝚌𝚑\_𝚜𝚒𝚖

𝙸𝙽𝙵𝙾: [𝚅𝚒𝚟𝚊𝚍𝚘 𝟷𝟸-𝟻𝟼𝟾𝟸] 𝙻𝚊𝚞𝚗𝚌𝚑𝚒𝚗𝚐 𝚋𝚎𝚑𝚊𝚟𝚒𝚘𝚛𝚊𝚕 𝚜𝚒𝚖𝚞𝚕𝚊𝚝𝚒𝚘𝚗 𝚒𝚗 '/𝚑𝚘𝚖𝚎/𝚜𝚝𝚞𝚍𝚎𝚗𝚝/𝙻𝚊𝚋𝚘𝟸/𝙻𝚊𝚋𝚘𝟸.𝚜𝚒𝚖/𝚜𝚒𝚖\_𝟷/𝚋𝚎𝚑𝚊𝚟/𝚡𝚜𝚒𝚖'

𝙸𝙽𝙵𝙾: [𝚂𝙸𝙼-𝚞𝚝𝚒𝚕𝚜-𝟻𝟷] 𝚂𝚒𝚖𝚞𝚕𝚊𝚝𝚒𝚘𝚗 𝚘𝚋𝚓𝚎𝚌𝚝 𝚒𝚜 '𝚜𝚒𝚖\_𝟷'

𝙸𝙽𝙵𝙾: [𝚂𝙸𝙼-𝚞𝚝𝚒𝚕𝚜-𝟻𝟺] 𝙸𝚗𝚜𝚙𝚎𝚌𝚝𝚒𝚗𝚐 𝚍𝚎𝚜𝚒𝚐𝚗 𝚜𝚘𝚞𝚛𝚌𝚎 𝚏𝚒𝚕𝚎𝚜 𝚏𝚘𝚛 '𝙰𝙻𝚄\_𝚝𝚋' 𝚒𝚗 𝚏𝚒𝚕𝚎𝚜𝚎𝚝 '𝚜𝚒𝚖\_𝟷'...

𝙸𝙽𝙵𝙾: [𝚄𝚂𝙵-𝚇𝚂𝚒𝚖-𝟿𝟽] 𝙵𝚒𝚗𝚍𝚒𝚗𝚐 𝚐𝚕𝚘𝚋𝚊𝚕 𝚒𝚗𝚌𝚕𝚞𝚍𝚎 𝚏𝚒𝚕𝚎𝚜...

𝙸𝙽𝙵𝙾: [𝚄𝚂𝙵-𝚇𝚂𝚒𝚖-𝟷00] 𝙵𝚎𝚝𝚌𝚑𝚒𝚗𝚐 𝚍𝚎𝚜𝚒𝚐𝚗 𝚏𝚒𝚕𝚎𝚜 𝚏𝚛𝚘𝚖 '𝚜𝚘𝚞𝚛𝚌𝚎𝚜\_𝟷'...(𝚝𝚑𝚒𝚜 𝚖𝚊𝚢 𝚝𝚊𝚔𝚎 𝚊 𝚠𝚑𝚒𝚕𝚎)...

𝙸𝙽𝙵𝙾: [𝚄𝚂𝙵-𝚇𝚂𝚒𝚖-𝟷0𝟷] 𝙵𝚎𝚝𝚌𝚑𝚒𝚗𝚐 𝚍𝚎𝚜𝚒𝚐𝚗 𝚏𝚒𝚕𝚎𝚜 𝚏𝚛𝚘𝚖 '𝚜𝚒𝚖\_𝟷'...

𝙸𝙽𝙵𝙾: [𝚄𝚂𝙵-𝚇𝚂𝚒𝚖-𝟸] 𝚇𝚂𝚒𝚖::𝙲𝚘𝚖𝚙𝚒𝚕𝚎 𝚍𝚎𝚜𝚒𝚐𝚗

𝙸𝙽𝙵𝙾: [𝚄𝚂𝙵-𝚇𝚂𝚒𝚖-𝟼𝟷] 𝙴𝚡𝚎𝚌𝚞𝚝𝚒𝚗𝚐 '𝙲𝙾𝙼𝙿𝙸𝙻𝙴 𝚊𝚗𝚍 𝙰𝙽𝙰𝙻𝚈𝚉𝙴' 𝚜𝚝𝚎𝚙 𝚒𝚗 '/𝚑𝚘𝚖𝚎/𝚜𝚝𝚞𝚍𝚎𝚗𝚝/𝙻𝚊𝚋𝚘𝟸/𝙻𝚊𝚋𝚘𝟸.𝚜𝚒𝚖/𝚜𝚒𝚖\_𝟷/𝚋𝚎𝚑𝚊𝚟/𝚡𝚜𝚒𝚖'

𝚡𝚟𝚑𝚍𝚕 --𝚒𝚗𝚌𝚛 --𝚛𝚎𝚕𝚊𝚡 -𝚙𝚛𝚓 𝙰𝙻𝚄\_𝚝𝚋\_𝚟𝚑𝚍𝚕.𝚙𝚛𝚓

𝙸𝙽𝙵𝙾: [𝚅𝚁𝙵𝙲 𝟷0-𝟷𝟼𝟹] 𝙰𝚗𝚊𝚕𝚢𝚣𝚒𝚗𝚐 𝚅𝙷𝙳𝙻 𝚏𝚒𝚕𝚎 "/𝚑𝚘𝚖𝚎/𝚜𝚝𝚞𝚍𝚎𝚗𝚝/𝙻𝚊𝚋𝚘𝟸/𝙻𝚊𝚋𝚘𝟸.𝚜𝚛𝚌𝚜/𝚜𝚘𝚞𝚛𝚌𝚎𝚜\_𝟷/𝚒𝚖𝚙𝚘𝚛𝚝𝚜/𝙰𝙻𝚄𝟾𝚋𝚒𝚝.𝚟𝚑𝚍" 𝚒𝚗𝚝𝚘 𝚕𝚒𝚋𝚛𝚊𝚛𝚢 𝚡𝚒𝚕\_𝚍𝚎𝚏𝚊𝚞𝚕𝚝𝚕𝚒𝚋

𝙸𝙽𝙵𝙾: [𝚅𝚁𝙵𝙲 𝟷0-𝟹0𝟽] 𝚊𝚗𝚊𝚕𝚢𝚣𝚒𝚗𝚐 𝚎𝚗𝚝𝚒𝚝𝚢 𝙰𝙻𝚄𝟾𝚋𝚒𝚝

𝙸𝙽𝙵𝙾: [𝚄𝚂𝙵-𝚇𝚂𝚒𝚖-𝟼𝟿] '𝚌𝚘𝚖𝚙𝚒𝚕𝚎' 𝚜𝚝𝚎𝚙 𝚏𝚒𝚗𝚒𝚜𝚑𝚎𝚍 𝚒𝚗 '𝟷' 𝚜𝚎𝚌𝚘𝚗𝚍𝚜

𝙸𝙽𝙵𝙾: [𝚅𝚒𝚟𝚊𝚍𝚘 𝟷𝟸-𝟻𝟼𝟾𝟸] 𝙻𝚊𝚞𝚗𝚌𝚑𝚒𝚗𝚐 𝚋𝚎𝚑𝚊𝚟𝚒𝚘𝚛𝚊𝚕 𝚜𝚒𝚖𝚞𝚕𝚊𝚝𝚒𝚘𝚗 𝚒𝚗 '/𝚑𝚘𝚖𝚎/𝚜𝚝𝚞𝚍𝚎𝚗𝚝/𝙻𝚊𝚋𝚘𝟸/𝙻𝚊𝚋𝚘𝟸.𝚜𝚒𝚖/𝚜𝚒𝚖\_𝟷/𝚋𝚎𝚑𝚊𝚟/𝚡𝚜𝚒𝚖'

𝙸𝙽𝙵𝙾: [𝚄𝚂𝙵-𝚇𝚂𝚒𝚖-𝟹] 𝚇𝚂𝚒𝚖::𝙴𝚕𝚊𝚋𝚘𝚛𝚊𝚝𝚎 𝚍𝚎𝚜𝚒𝚐𝚗

𝙸𝙽𝙵𝙾: [𝚄𝚂𝙵-𝚇𝚂𝚒𝚖-𝟼𝟷] 𝙴𝚡𝚎𝚌𝚞𝚝𝚒𝚗𝚐 '𝙴𝙻𝙰𝙱𝙾𝚁𝙰𝚃𝙴' 𝚜𝚝𝚎𝚙 𝚒𝚗 '/𝚑𝚘𝚖𝚎/𝚜𝚝𝚞𝚍𝚎𝚗𝚝/𝙻𝚊𝚋𝚘𝟸/𝙻𝚊𝚋𝚘𝟸.𝚜𝚒𝚖/𝚜𝚒𝚖\_𝟷/𝚋𝚎𝚑𝚊𝚟/𝚡𝚜𝚒𝚖'

𝚅𝚒𝚟𝚊𝚍𝚘 𝚂𝚒𝚖𝚞𝚕𝚊𝚝𝚘𝚛 𝟸0𝟷𝟽.𝟺

𝙲𝚘𝚙𝚢𝚛𝚒𝚐𝚑𝚝 𝟷𝟿𝟾𝟼-𝟷𝟿𝟿𝟿, 𝟸00𝟷-𝟸0𝟷𝟼 𝚇𝚒𝚕𝚒𝚗𝚡, 𝙸𝚗𝚌. 𝙰𝚕𝚕 𝚁𝚒𝚐𝚑𝚝𝚜 𝚁𝚎𝚜𝚎𝚛𝚟𝚎𝚍.

𝚁𝚞𝚗𝚗𝚒𝚗𝚐: /𝚘𝚙𝚝/𝚇𝚒𝚕𝚒𝚗𝚡/𝚅𝚒𝚟𝚊𝚍𝚘/𝟸0𝟷𝟽.𝟺/𝚋𝚒𝚗/𝚞𝚗𝚠𝚛𝚊𝚙𝚙𝚎𝚍/𝚕𝚗𝚡𝟼𝟺.𝚘/𝚡𝚎𝚕𝚊𝚋 -𝚠𝚝𝚘 0𝚏𝟼𝚊𝚌𝚌𝟾𝟿𝚊𝟸𝚎𝟿𝟺00𝟸𝚊𝚍𝚋𝟿𝟷𝟾𝚍0𝟿𝟸𝟿𝚍𝚌𝚎𝟺𝟾 --𝚒𝚗𝚌𝚛 --𝚍𝚎𝚋𝚞𝚐 𝚝𝚢𝚙𝚒𝚌𝚊𝚕 --𝚛𝚎𝚕𝚊𝚡 --𝚖𝚝 𝟾 -𝙻 𝚡𝚒𝚕\_𝚍𝚎𝚏𝚊𝚞𝚕𝚝𝚕𝚒𝚋 -𝙻 𝚜𝚎𝚌𝚞𝚛𝚎𝚒𝚙 --𝚜𝚗𝚊𝚙𝚜𝚑𝚘𝚝 𝙰𝙻𝚄\_𝚝𝚋\_𝚋𝚎𝚑𝚊𝚟 𝚡𝚒𝚕\_𝚍𝚎𝚏𝚊𝚞𝚕𝚝𝚕𝚒𝚋.𝙰𝙻𝚄\_𝚝𝚋 -𝚕𝚘𝚐 𝚎𝚕𝚊𝚋𝚘𝚛𝚊𝚝𝚎.𝚕𝚘𝚐

𝚄𝚜𝚒𝚗𝚐 𝟾 𝚜𝚕𝚊𝚟𝚎 𝚝𝚑𝚛𝚎𝚊𝚍𝚜.

𝚂𝚝𝚊𝚛𝚝𝚒𝚗𝚐 𝚜𝚝𝚊𝚝𝚒𝚌 𝚎𝚕𝚊𝚋𝚘𝚛𝚊𝚝𝚒𝚘𝚗

𝙲𝚘𝚖𝚙𝚕𝚎𝚝𝚎𝚍 𝚜𝚝𝚊𝚝𝚒𝚌 𝚎𝚕𝚊𝚋𝚘𝚛𝚊𝚝𝚒𝚘𝚗

𝚂𝚝𝚊𝚛𝚝𝚒𝚗𝚐 𝚜𝚒𝚖𝚞𝚕𝚊𝚝𝚒𝚘𝚗 𝚍𝚊𝚝𝚊 𝚏𝚕𝚘𝚠 𝚊𝚗𝚊𝚕𝚢𝚜𝚒𝚜

𝙲𝚘𝚖𝚙𝚕𝚎𝚝𝚎𝚍 𝚜𝚒𝚖𝚞𝚕𝚊𝚝𝚒𝚘𝚗 𝚍𝚊𝚝𝚊 𝚏𝚕𝚘𝚠 𝚊𝚗𝚊𝚕𝚢𝚜𝚒𝚜

𝚃𝚒𝚖𝚎 𝚁𝚎𝚜𝚘𝚕𝚞𝚝𝚒𝚘𝚗 𝚏𝚘𝚛 𝚜𝚒𝚖𝚞𝚕𝚊𝚝𝚒𝚘𝚗 𝚒𝚜 𝟷𝚙𝚜

𝙲𝚘𝚖𝚙𝚒𝚕𝚒𝚗𝚐 𝚙𝚊𝚌𝚔𝚊𝚐𝚎 𝚜𝚝𝚍.𝚜𝚝𝚊𝚗𝚍𝚊𝚛𝚍

𝙲𝚘𝚖𝚙𝚒𝚕𝚒𝚗𝚐 𝚙𝚊𝚌𝚔𝚊𝚐𝚎 𝚜𝚝𝚍.𝚝𝚎𝚡𝚝𝚒𝚘

𝙲𝚘𝚖𝚙𝚒𝚕𝚒𝚗𝚐 𝚙𝚊𝚌𝚔𝚊𝚐𝚎 𝚒𝚎𝚎𝚎.𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝟷𝟷𝟼𝟺

𝙲𝚘𝚖𝚙𝚒𝚕𝚒𝚗𝚐 𝚙𝚊𝚌𝚔𝚊𝚐𝚎 𝚒𝚎𝚎𝚎.𝚗𝚞𝚖𝚎𝚛𝚒𝚌\_𝚜𝚝𝚍

𝙲𝚘𝚖𝚙𝚒𝚕𝚒𝚗𝚐 𝚙𝚊𝚌𝚔𝚊𝚐𝚎 𝚒𝚎𝚎𝚎.𝚜𝚝𝚍\_𝚕𝚘𝚐𝚒𝚌\_𝚝𝚎𝚡𝚝𝚒𝚘

𝙲𝚘𝚖𝚙𝚒𝚕𝚒𝚗𝚐 𝚙𝚊𝚌𝚔𝚊𝚐𝚎 𝚡𝚒𝚕\_𝚍𝚎𝚏𝚊𝚞𝚕𝚝𝚕𝚒𝚋.𝚙𝚛𝚘𝚌𝚎𝚜𝚜𝚘𝚛\_𝚙𝚔𝚐

𝙲𝚘𝚖𝚙𝚒𝚕𝚒𝚗𝚐 𝚊𝚛𝚌𝚑𝚒𝚝𝚎𝚌𝚝𝚞𝚛𝚎 𝚕𝚍𝚍𝟷 𝚘𝚏 𝚎𝚗𝚝𝚒𝚝𝚢 𝚡𝚒𝚕\_𝚍𝚎𝚏𝚊𝚞𝚕𝚝𝚕𝚒𝚋.𝙵𝙰𝟷𝙱 [𝚏𝚊𝟷𝚋\_𝚍𝚎𝚏𝚊𝚞𝚕𝚝]

𝙲𝚘𝚖𝚙𝚒𝚕𝚒𝚗𝚐 𝚊𝚛𝚌𝚑𝚒𝚝𝚎𝚌𝚝𝚞𝚛𝚎 𝚕𝚍𝚍𝟷 𝚘𝚏 𝚎𝚗𝚝𝚒𝚝𝚢 𝚡𝚒𝚕\_𝚍𝚎𝚏𝚊𝚞𝚕𝚝𝚕𝚒𝚋.𝙰𝙳𝙳 [\𝙰𝙳𝙳(𝚌\_𝚍𝚊𝚝𝚊\_𝚠𝚒𝚍𝚝𝚑=𝟾)\]

𝙲𝚘𝚖𝚙𝚒𝚕𝚒𝚗𝚐 𝚊𝚛𝚌𝚑𝚒𝚝𝚎𝚌𝚝𝚞𝚛𝚎 𝚋𝚎𝚑𝚊𝚟𝚒𝚘𝚛𝚊𝚕 𝚘𝚏 𝚎𝚗𝚝𝚒𝚝𝚢 𝚡𝚒𝚕\_𝚍𝚎𝚏𝚊𝚞𝚕𝚝𝚕𝚒𝚋.𝙰𝙻𝚄𝟾𝚋𝚒𝚝 [𝚊𝚕𝚞𝟾𝚋𝚒𝚝\_𝚍𝚎𝚏𝚊𝚞𝚕𝚝]

𝙲𝚘𝚖𝚙𝚒𝚕𝚒𝚗𝚐 𝚊𝚛𝚌𝚑𝚒𝚝𝚎𝚌𝚝𝚞𝚛𝚎 𝚋𝚎𝚑𝚊𝚟𝚒𝚘𝚛𝚊𝚕 𝚘𝚏 𝚎𝚗𝚝𝚒𝚝𝚢 𝚡𝚒𝚕\_𝚍𝚎𝚏𝚊𝚞𝚕𝚝𝚕𝚒𝚋.𝚊𝚕𝚞\_𝚝𝚋

𝙱𝚞𝚒𝚕𝚝 𝚜𝚒𝚖𝚞𝚕𝚊𝚝𝚒𝚘𝚗 𝚜𝚗𝚊𝚙𝚜𝚑𝚘𝚝 𝙰𝙻𝚄\_𝚝𝚋\_𝚋𝚎𝚑𝚊𝚟

𝙸𝙽𝙵𝙾: [𝚄𝚂𝙵-𝚇𝚂𝚒𝚖-𝟼𝟿] '𝚎𝚕𝚊𝚋𝚘𝚛𝚊𝚝𝚎' 𝚜𝚝𝚎𝚙 𝚏𝚒𝚗𝚒𝚜𝚑𝚎𝚍 𝚒𝚗 '𝟷' 𝚜𝚎𝚌𝚘𝚗𝚍𝚜

𝚅𝚒𝚟𝚊𝚍𝚘 𝚂𝚒𝚖𝚞𝚕𝚊𝚝𝚘𝚛 𝟸0𝟷𝟽.𝟺

𝚃𝚒𝚖𝚎 𝚛𝚎𝚜𝚘𝚕𝚞𝚝𝚒𝚘𝚗 𝚒𝚜 𝟷 𝚙𝚜

𝚂𝚄𝙲𝙲𝙴𝚂𝚂: 𝙽𝙾𝚃.

𝚂𝚄𝙲𝙲𝙴𝚂𝚂: 𝙰𝙽𝙳.

𝚂𝚄𝙲𝙲𝙴𝚂𝚂: 𝙾𝚁.

𝚂𝚄𝙲𝙲𝙴𝚂𝚂: 𝚇𝙾𝚁.

𝚂𝚄𝙲𝙲𝙴𝚂𝚂: 𝙰𝙳𝙳 𝟷.

𝚂𝚄𝙲𝙲𝙴𝚂𝚂: 𝙰𝙳𝙳 𝟸.

𝚂𝚄𝙲𝙲𝙴𝚂𝚂: 𝙰𝙳𝙳 𝟹.

𝚂𝚄𝙲𝙲𝙴𝚂𝚂: 𝚂𝚄𝙱 𝟷.

𝚂𝚄𝙲𝙲𝙴𝚂𝚂: 𝚂𝚄𝙱 𝟸.

𝚂𝚄𝙲𝙲𝙴𝚂𝚂: 𝙲𝙼𝙿 𝚎𝚚𝚞𝚊𝚕.

𝚂𝚄𝙲𝙲𝙴𝚂𝚂: 𝙲𝙼𝙿 𝚜𝚖𝚊𝚕𝚕𝚎𝚛.

𝚂𝚄𝙲𝙲𝙴𝚂𝚂: 𝙲𝙼𝙿 𝚐𝚛𝚎𝚊𝚝𝚎𝚛.

𝚂𝚄𝙲𝙲𝙴𝚂𝚂: 𝚁𝚁 𝟷.

𝚂𝚄𝙲𝙲𝙴𝚂𝚂: 𝚁𝚁 𝟸.

𝚂𝚄𝙲𝙲𝙴𝚂𝚂: 𝚁𝙻 𝟷.

𝚂𝚄𝙲𝙲𝙴𝚂𝚂: 𝚁𝙻 𝟸.

𝚂𝚄𝙲𝙲𝙴𝚂𝚂: 𝚂𝚆𝙰𝙿.

𝙽𝚘𝚝𝚎: 𝚂𝙸𝙼𝚄𝙻𝙰𝚃𝙸𝙾𝙽 𝙴𝙽𝙳𝙴𝙳

𝚃𝚒𝚖𝚎: 𝟸0𝟽 𝚗𝚜 𝙸𝚝𝚎𝚛𝚊𝚝𝚒𝚘𝚗: 0 𝙿𝚛𝚘𝚌𝚎𝚜𝚜: /𝙰𝙻𝚄\_𝚝𝚋/𝚂𝚃𝙸𝙼\_𝙿𝚁𝙾𝙲 𝙵𝚒𝚕𝚎: /𝚑𝚘𝚖𝚎/𝚜𝚝𝚞𝚍𝚎𝚗𝚝/𝙻𝚊𝚋𝚘𝟸/𝙻𝚊𝚋𝚘𝟸.𝚜𝚛𝚌𝚜/𝚜𝚒𝚖\_𝟷/𝚒𝚖𝚙𝚘𝚛𝚝𝚜/𝚜𝚘𝚞𝚛𝚌𝚎𝚜/𝙰𝙻𝚄\_𝚝𝚋.𝚟𝚑𝚍

# Besluit

-wat is er gerealiseerd

-wat niet -> denk verder

-oplossingen

# Vragen