# Lecture – 12 Date: 13.09.2014

- The Signal Flow Graph

# Signal Flow Graph

**Q:** Using individual device scattering parameters to analyze a complex microwave network results in a lot of **messy** math! Isn't there an **easier** way?

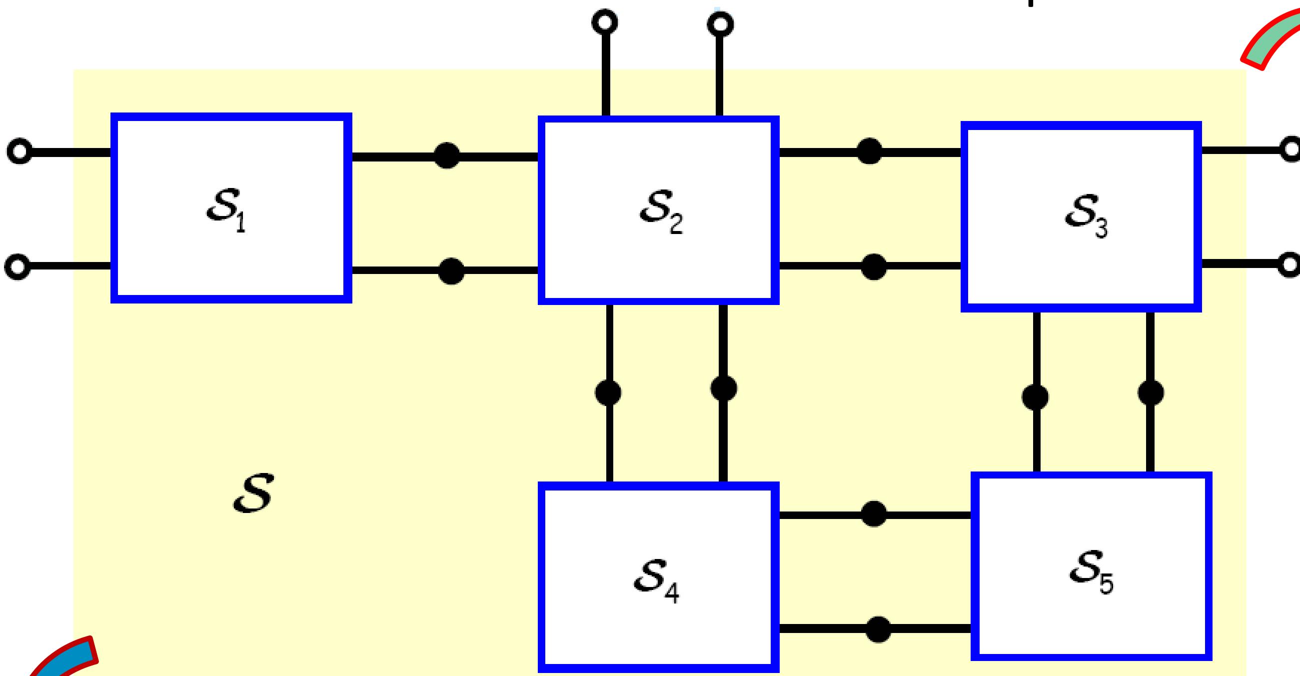
**A:** Yes! We can represent a microwave network with its **signal flow graph** and then decompose this graph using a standard set of rules → results into simpler analysis.

It provides a sort of a **graphical** way to do algebra!

Signal Flow Graph (SFG) can also help us understand the fundamental **physical behavior** of a network or device. It can even help us **approximate** the network in a way that makes it simpler to analyze and/or design!

# Signal Flow Graph (contd.)

- To understand the significance of SFG, let us consider a complex **3-port** microwave network constructed of **5** simpler microwave devices




$S_n$ is the **scattering matrix** of each device, and $S$ is the **overall** scattering matrix of the **entire** 3-port network

The S-parameter (S) of the whole network can be obtained from the knowledge of S-parameter of individual devices

**Tedious Algebra!**

**Alternative is SFG based solution!**

# Signal Flow Graph (contd.)

## Signal flow graphs are helpful in three ways!

**Way 1** – It provide us with a **graphical** means of **solving** large systems of simultaneous equations.

**Way 2** – We'll see that it can provide us with a **road map** of the wave **propagation paths** throughout a HF device or network. If we're paying attention, we can glean great **physical insight** as to the inner working of the device represented by the graph.

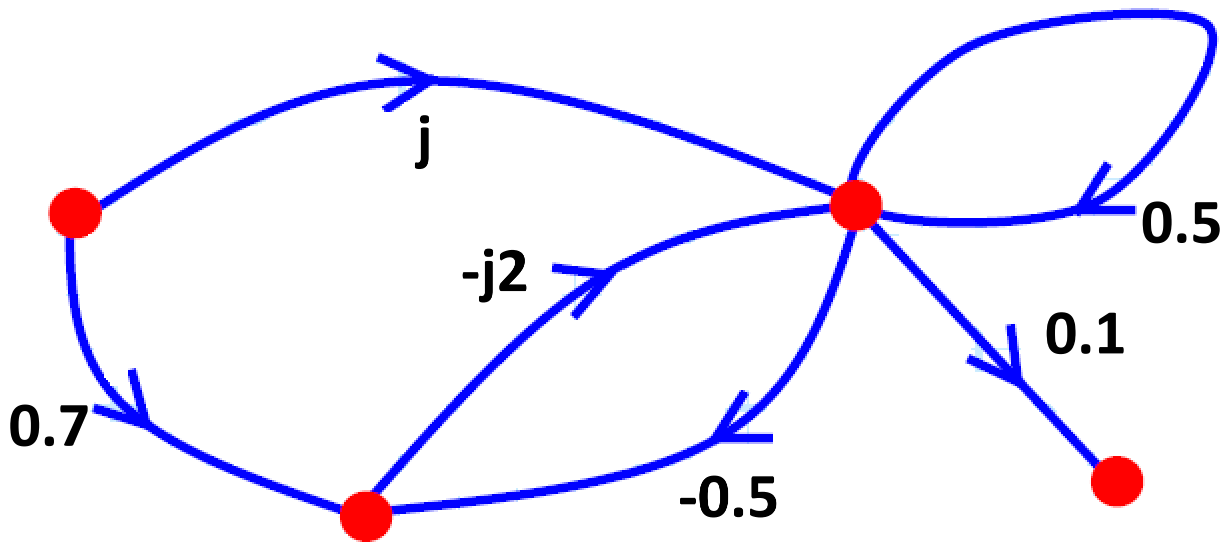
**Way 3** – It provide us with a quick and accurate method for **approximating** a network or device. We will find that we can often replace a rather complex graph with a much **simpler** one that is **almost** equivalent.

We find this to be very helpful when **designing** microwave components. From the analysis of these approximate graphs, we can often determine **design rules** or equations that are tractable, and allow us to design components with (near) optimal performance.

# Signal Flow Graph (contd.)

## Some definitions!

Every SFG consists of a set of **nodes**. These nodes are connected by **branches**, which are simply contours with a specified **direction**. Similarly, each branch has an associated complex **value**.




**Q:** What could this possibly have to do with **RF/microwave engineering**?

# Signal Flow Graph (contd.)

- In high frequency applications, each **port** of a device is represented by **two nodes**—the "a" node and the "b" node. The "a" node simply represents the value of the **normalized amplitude** of the wave incident on that port, evaluated **at** the plane of that port:

$$a_n = \frac{V_n^+(z_n = z_{nP})}{\sqrt{Z_{0n}}}$$

- Similarly, the "b" node simply represents the **normalized amplitude** of the wave **exiting** that port, evaluated **at** the plane of that port:

$$b_n = \frac{V_n^-(z_n = z_{nP})}{\sqrt{Z_{0n}}}$$

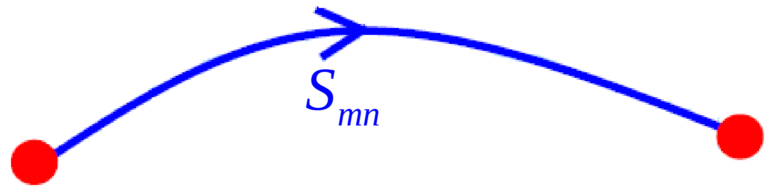
- Note then that the **total voltage** at a port is simply:

$$V_n(z_n = z_{nP}) = (a_n + b_n)\sqrt{Z_{0n}}$$

# Signal Flow Graph (contd.)

- The value of the **branch** connecting two nodes is simply the value of the **scattering parameter** relating these two voltage values.

$$a_n = \frac{V_n^+(z_n = z_{nP})}{\sqrt{Z_{0n}}}$$

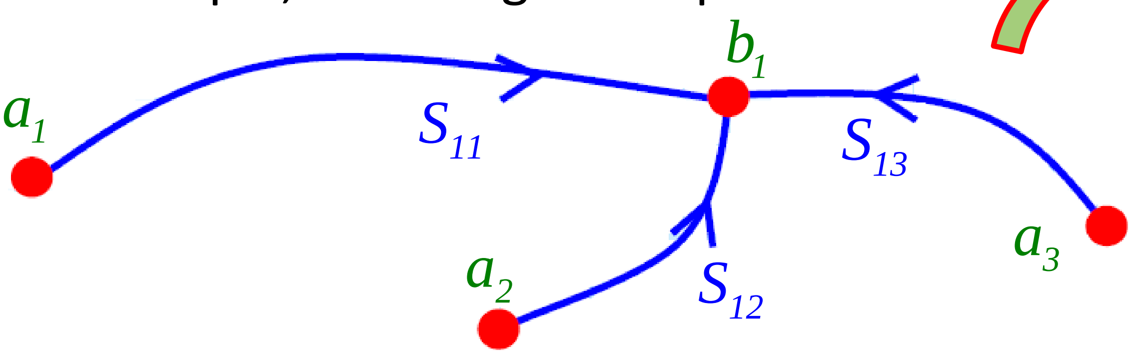



$$b_m = \frac{V_m^-(z_m = z_{mP})}{\sqrt{Z_{0m}}}$$

- The signal flow graph above is simply a **graphical** representation of the equation:

$$b_m = a_n S_{mn}$$

- Moreover, if **multiple** branches enter a node, then the voltage represented by that node is the **sum** of the values from each branch. For example, following SFG represents:




$$b_1 = S_{11}a_1 + S_{12}a_2 + S_{13}a_3$$

# Signal Flow Graph (contd.)

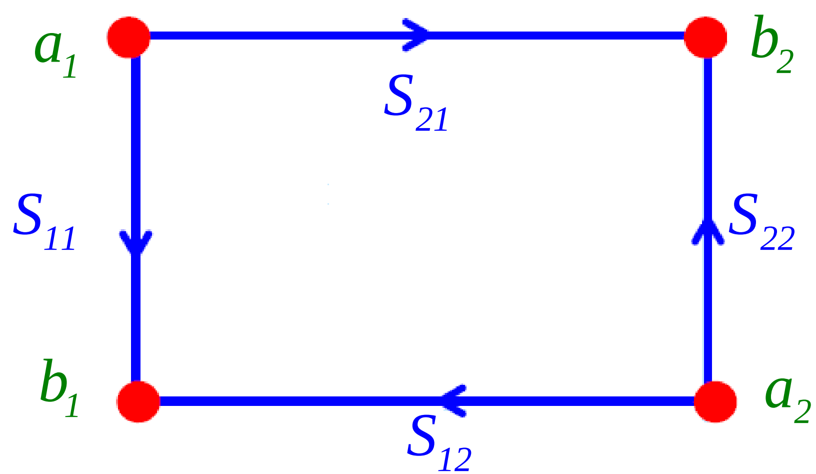
- Now, consider a **two-port device** with a scattering matrix **S:**

$$\mathbf{S} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}$$

**So that:**

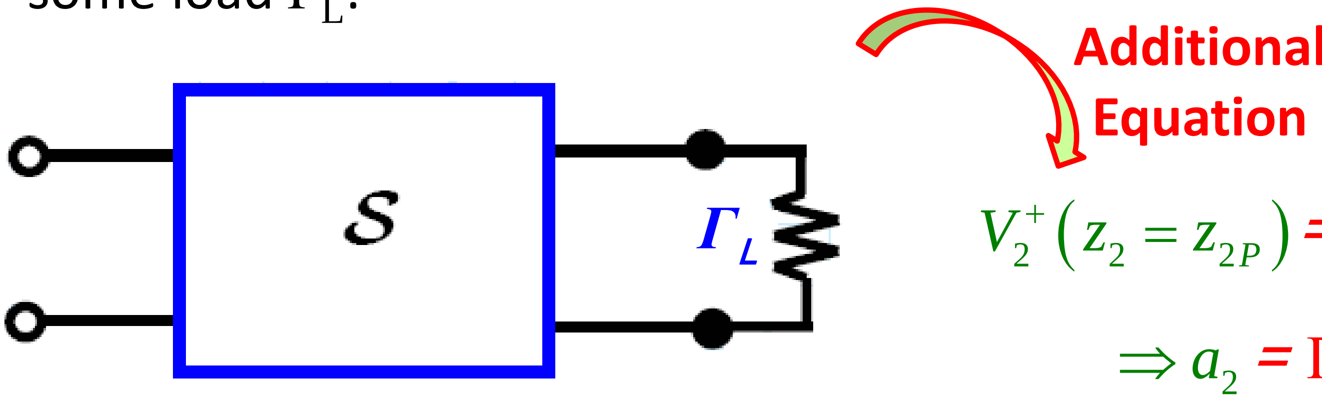
$$b_1 = S_{11}a_1 + S_{12}a_2 \qquad b_2 = S_{21}a_1 + S_{22}a_2$$

- We can then **graphically** represent a **two-port device** as:



# Signal Flow Graph (contd.)

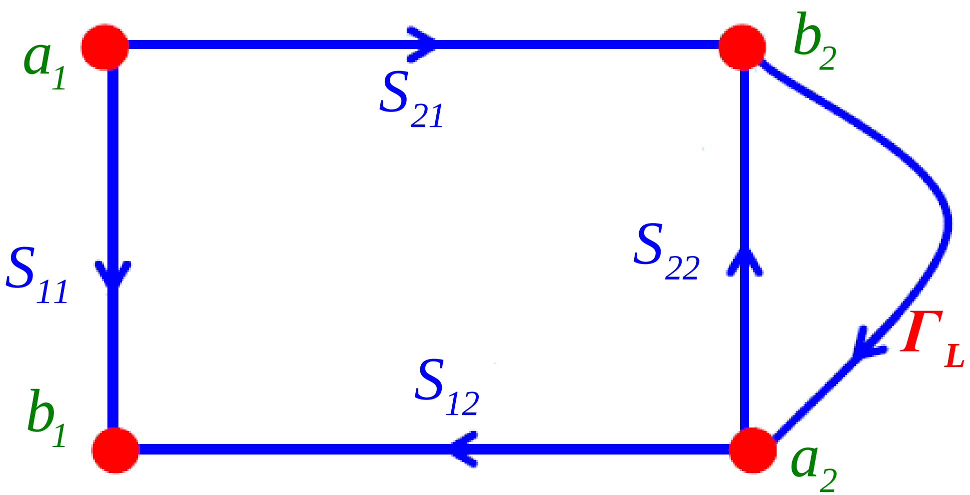
- Now, consider a two-port device where the second port is **terminated** by some load $\Gamma_L$:




$$V_2^+(z_2 = z_{2P}) = \Gamma_L V_2^-(z_2 = z_{2P})$$

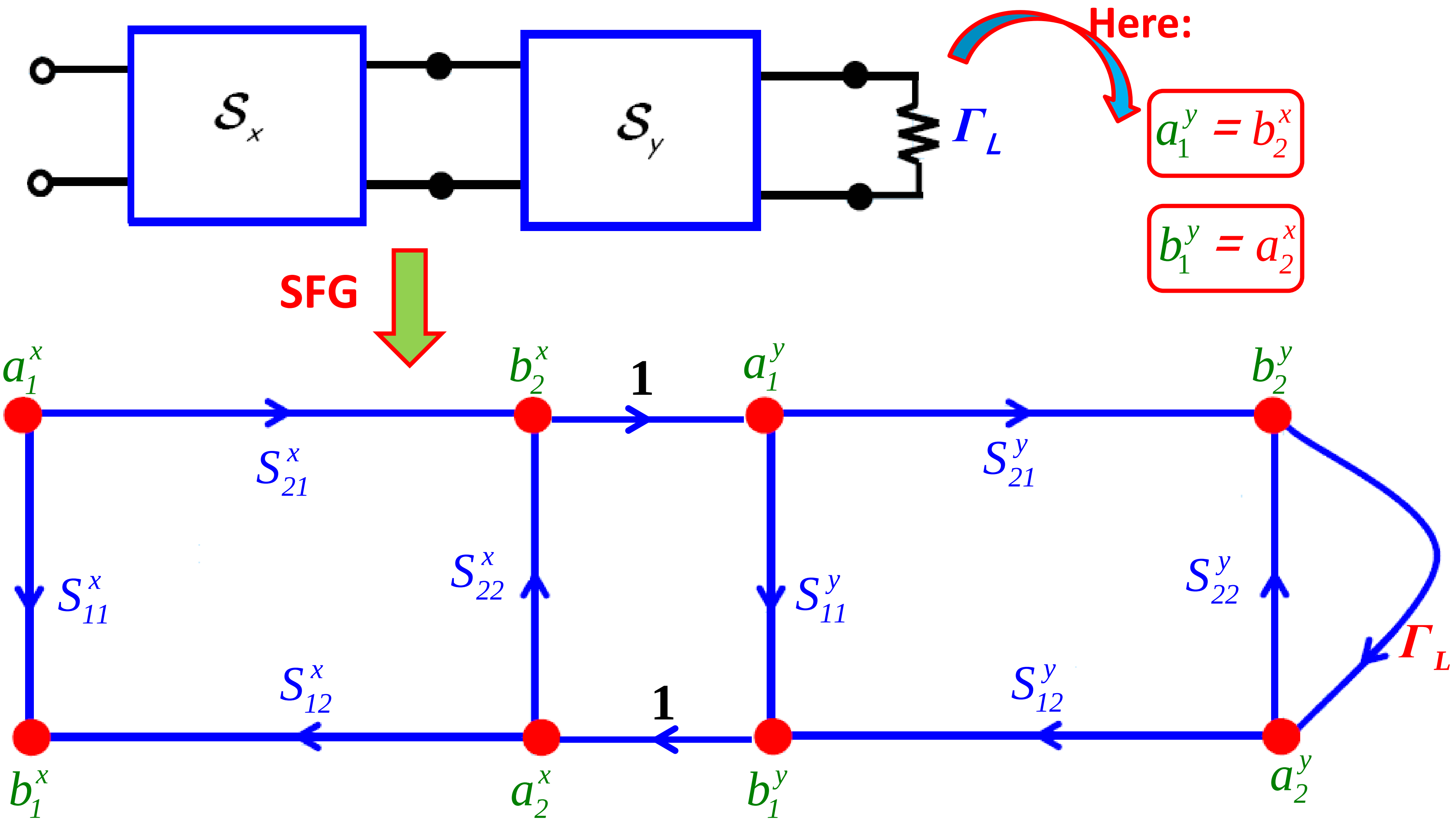
$$\Rightarrow a_2 = \Gamma_L b_2$$

- Therefore, the signal flow graph of this **terminated** network is:



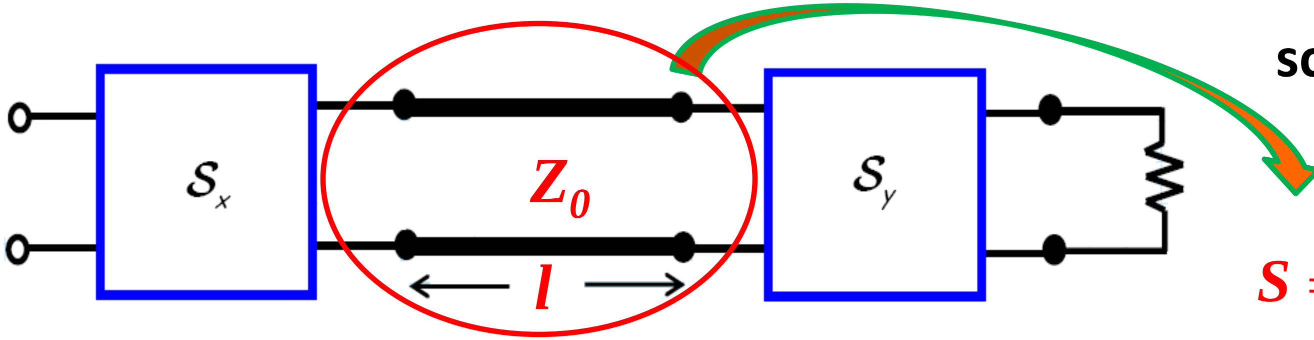
# Signal Flow Graph (contd.)

- Now consider cascading of **two different** two-port networks



# Signal Flow Graph (contd.)

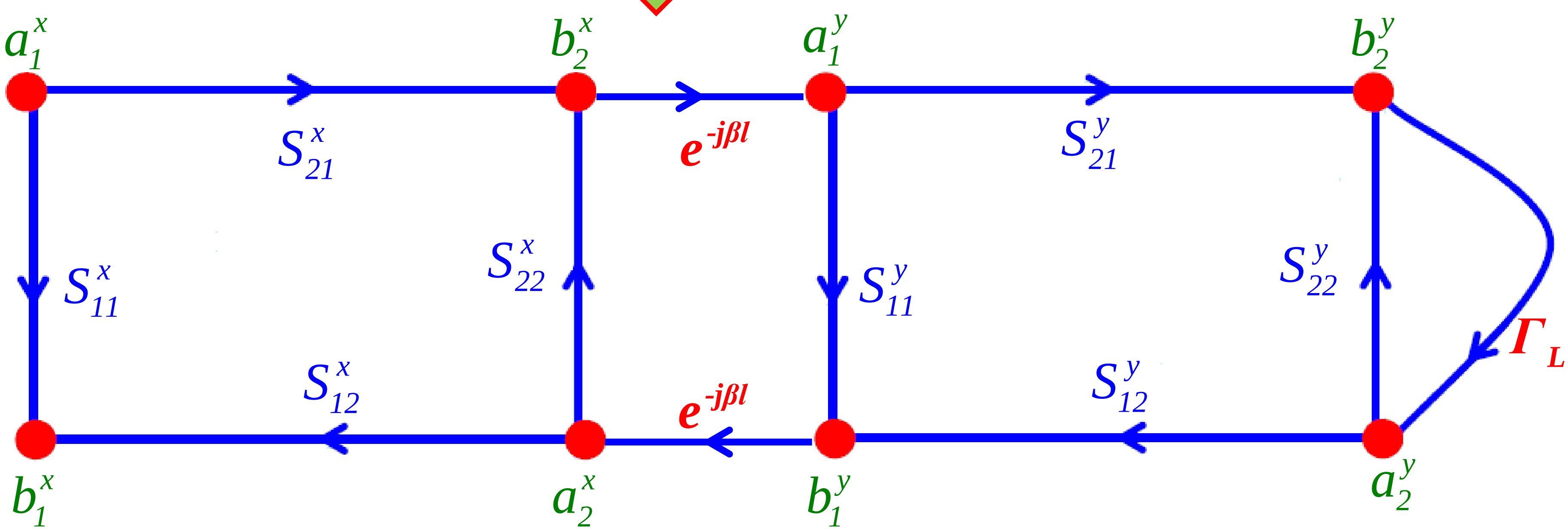
- Now consider networks connected with a **transmission line segment:**



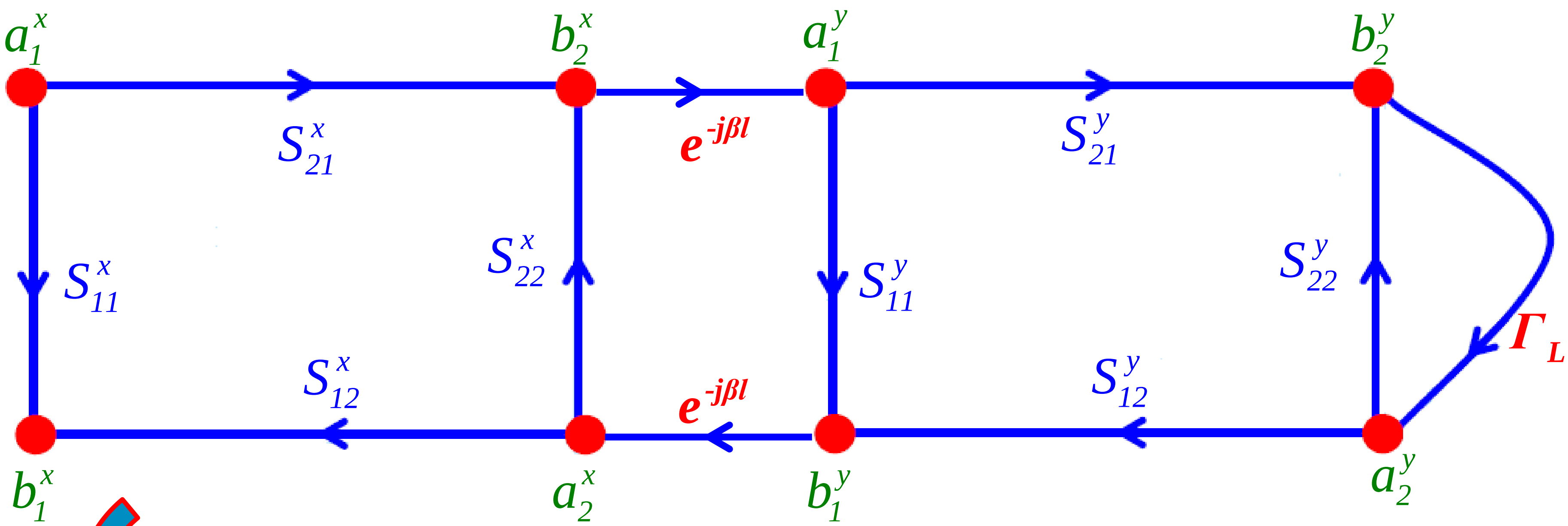

**scattering matrix**

$$S = \begin{bmatrix} 0 & e^{-j\beta l} \\ e^{-j\beta l} & 0 \end{bmatrix}$$

**SFG**



# Signal Flow Graph (contd.)




Note that there is **one** (and only one!) **independent variable** in this graphical representation (i.e., SFG) → $a_1^x$

This is the only node of the SFG that does **not** have any **incoming** branches. As a result, its value depends on **no other** node values in the SFG

**Independent nodes in the SFG are called sources!**

# Signal Flow Graph (contd.)

**Independent nodes in the SFG are called sources!**

- This makes sense physically (do **you** see why?)
- The node value $\mathbf{a_1^x}$ represents the complex amplitude of the wave **incident** on the one-port network. If this value is **zero**, then **no power** is incident on the network—the rest of the nodes (i.e., wave amplitudes) will be **zero**!

**Now, say we wish to determine, for example:**

**1.** The **reflection coefficient** $\Gamma_{in}$ of the one-port device

**2.** The **total current** at port 1 of second network (i.e., network y)

**3.** The **power absorbed** by the load at port 2 of the second (y) network.

# Signal Flow Graph (contd.)

- In the first case, we need to determine the value of dependent node ${b_1}^x$:

$$\Gamma_{in} = \frac{b_1^x}{a_1^x}$$

- For the second case, we must determine the value of wave amplitudes ${a_1}^y$ and ${b_1}^y$:

$$I_1^y = \frac{a_1^y - b_1^y}{\sqrt{Z_0}}$$

- For the third and final case, the values of nodes ${a_2}^y$ and ${b_2}^y$ are required:

$$P_{abs} = \frac{\left|b_2^y\right|^2 - \left|a_2^y\right|^2}{2}$$

How do we **determine** the values of these wave amplitude "nodes"?

solve the **simultaneous equations** that describe this network.

**Decompose (reduce)** the SFG!

# Signal Flow Graph (contd.)

- SFG **reduction** is a method for **simplifying** the **complex** paths of that SFG into a more **direct (but equivalent!)** form.
    - Reduction is really just a **graphical** method of **decoupling** the simultaneous equations that are **described** by the SFG.
- SFGs can be reduced by applying one of **four simple rules.**

**Q:** Can these rules be applied in **any order**?

**A: YES!** The rules can only be applied when/where the structure of the SFG allows. You must **search** the SFG for structures that allow a rule to be applied, and the SFG will then be (a little bit) reduced. You then search for the **next** valid structure where a rule can be applied. Eventually, the SFG will be **completely reduced**!

It's a bit like solving a **puzzle**. Every SFG is different, and so each requires a different reduction procedure. It requires a little **thought**, but with a little practice, the reduction procedure can be **easily** mastered → **You may find its kind of a fun! (TRUST ME)**

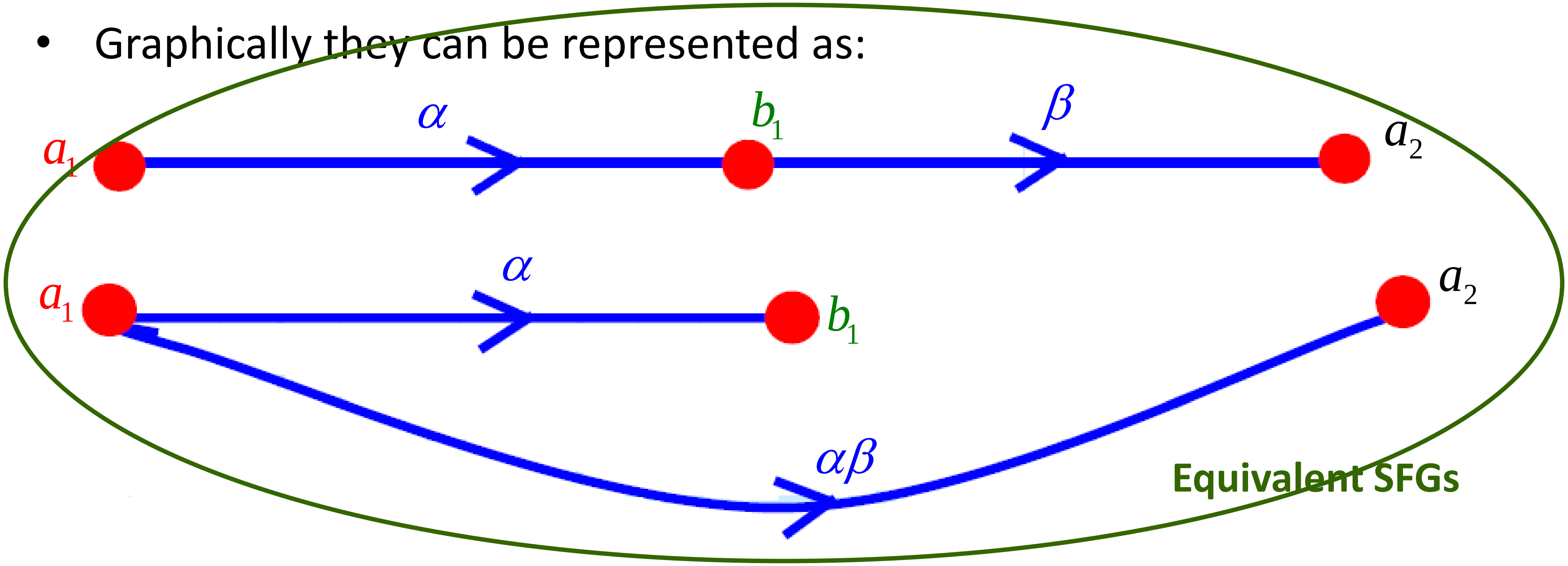
# Signal Flow Graph (contd.)

## Series Rule

- Consider these two complex equations: $b_1 = \alpha a_1$ $a_2 = \beta b_1$
- These two equations can combined to form an **equivalent set** of equations:

$$b_1 = \alpha a_1 \qquad a_2 = \beta b_1 = \beta(\alpha a_1) = \alpha\beta a_1$$

- Graphically they can be represented as:



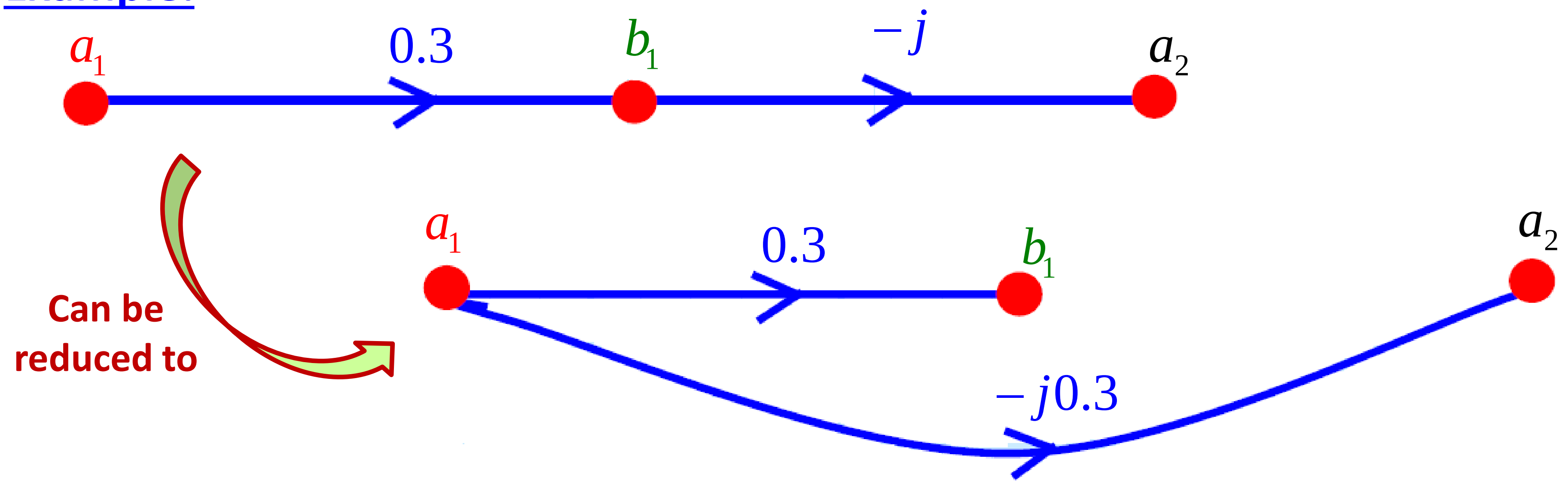
# Signal Flow Graph (contd.)

This last discussion leads us to our **first** SFG **reduction rule:**

**Rule 1 - Series Rule**

If a node has **one** (and only one!) incoming branch, and **one** (and only one!) outgoing branch, the node can be eliminated and the two branches can be combined, with the new branch having a value equal to the product of the original two.

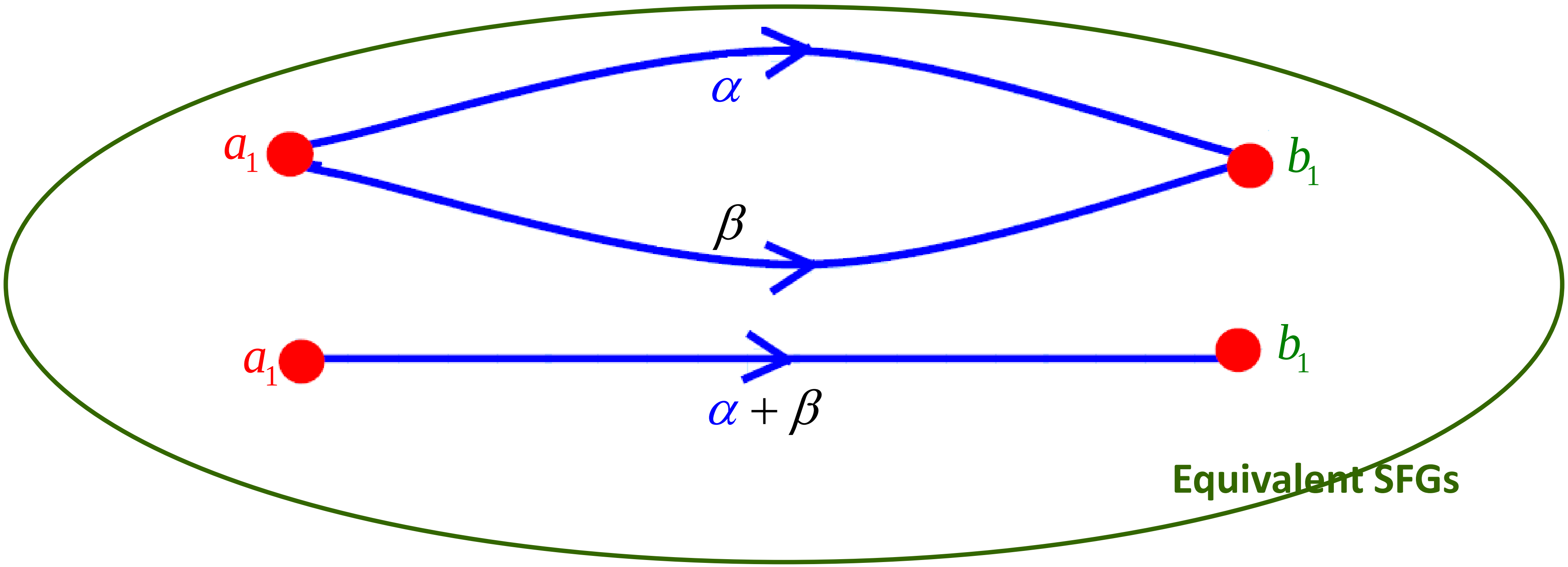
**Example:**



# Signal Flow Graph (contd.)

## Parallel Rule

- Consider these two complex equations: $b_1 = \alpha a_1 + \beta a_1$
- The equation can also be expressed as: $b_1 = (\alpha + \beta) a_1$
- These equations can be expressed in terms of SFG as:



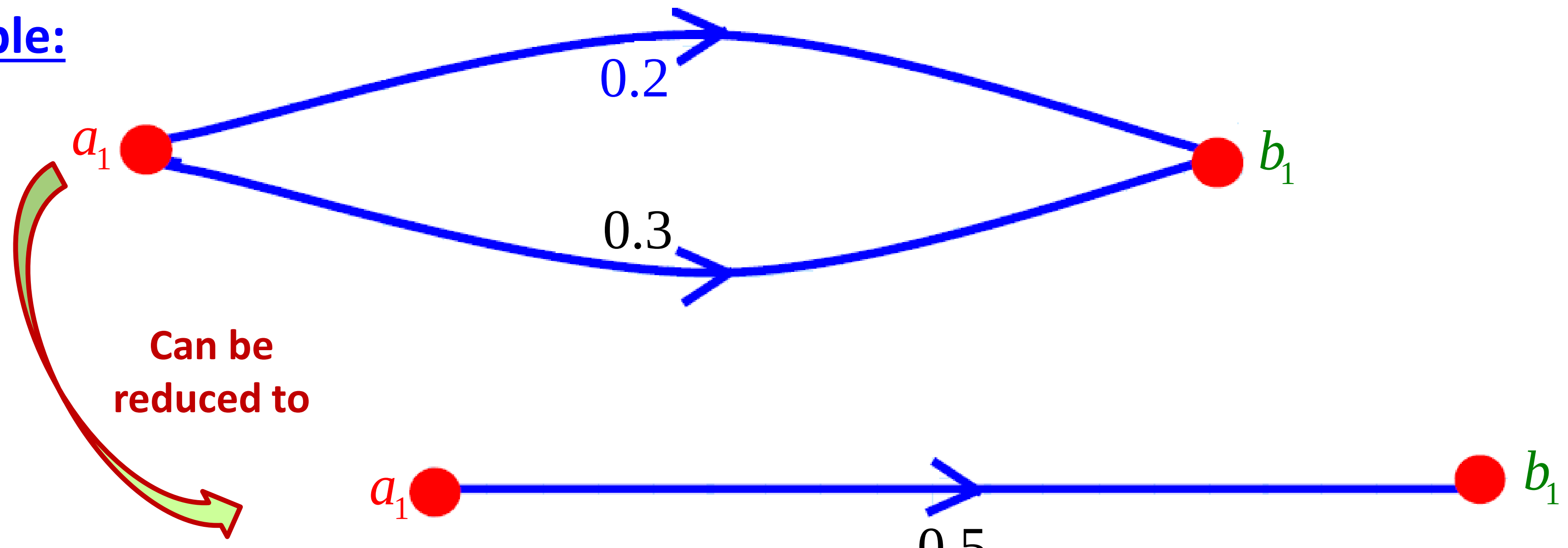
# Signal Flow Graph (contd.)

This last discussion leads us to our **second** SFG **reduction rule:**

**Rule 2 - Parallel Rule**

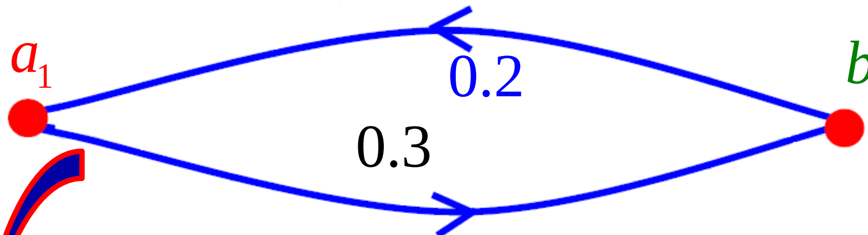
If two nodes are connected by parallel branches—and the branches have the **same direction**—the branches can be combined into a single branch, with a value equal to the **sum** of each two original branches.

**Example:**

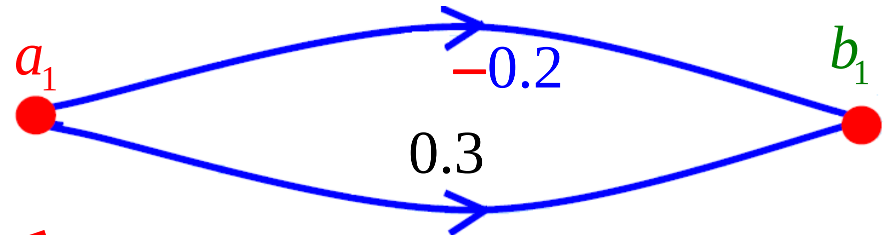


# Signal Flow Graph (contd.)

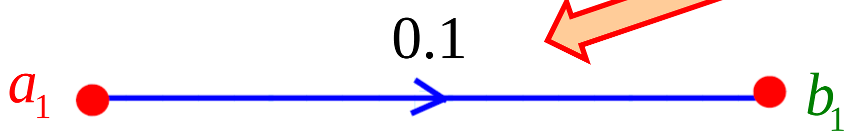
What about **this** signal flow graph?




Can it be transformed into

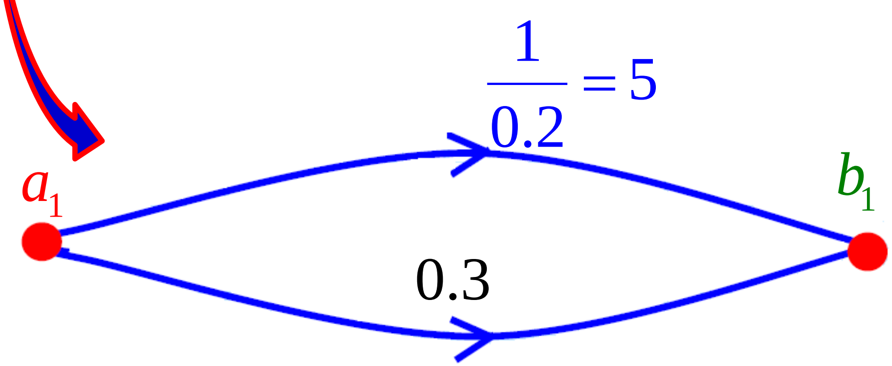



**So that**



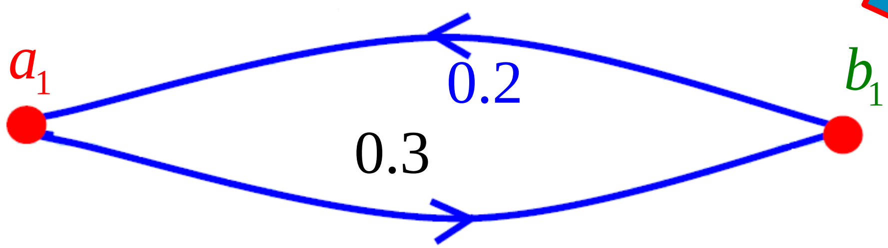

Absolutely not!
**NEVER DO THIS!!**

**How about:**




Absolutely not! **NEVER DO THIS EITHER!!**

# Signal Flow Graph (contd.)




Actually from this SFG we can only conclude that

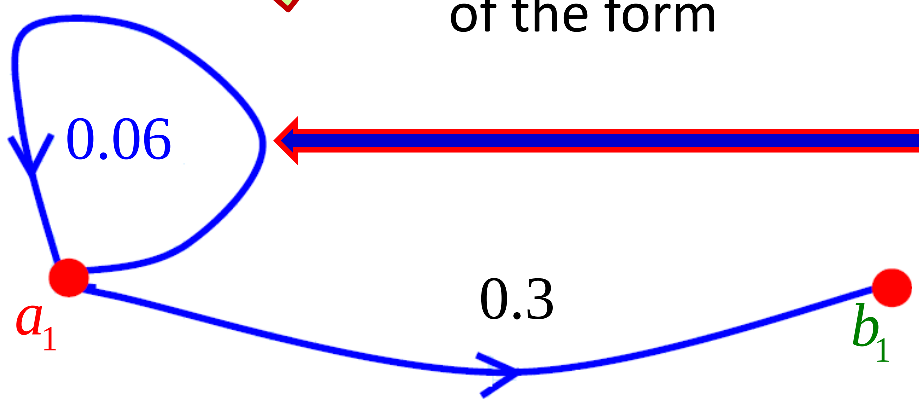
$b_1 = 0.3a_1$

$a_1 = 0.2b_1$

$a_1 = 0.06a_1$

$b_1 = 0.3a_1$

SFG can be of the form

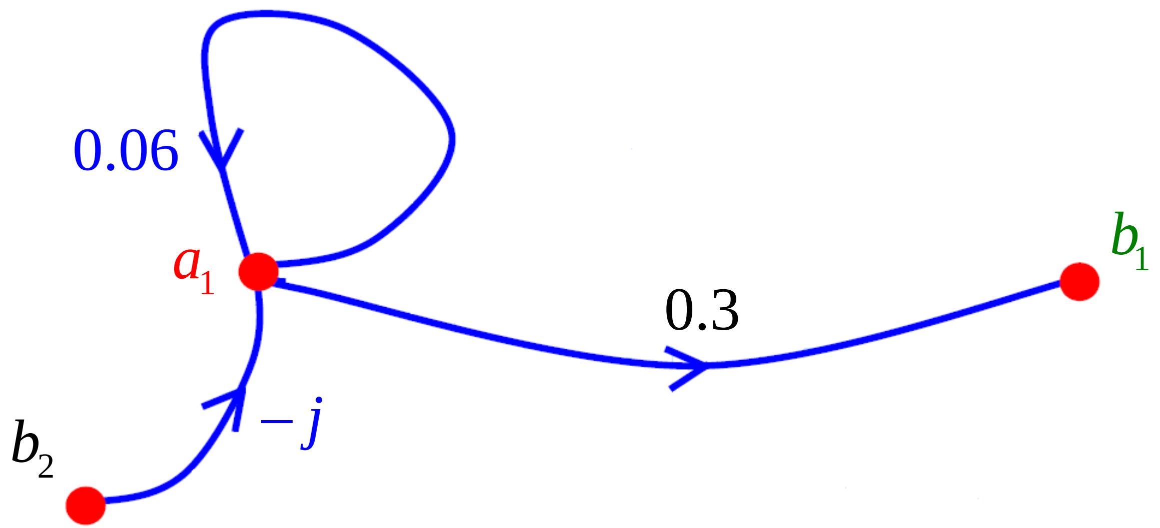



Branches that begin and end at the same node are called self-loops

In practical situations, self-loop node will **always** have at least **one other incoming branch**

# Signal Flow Graph (contd.)

## Practical example of node with self-loop:



# Signal Flow Graph (contd.)

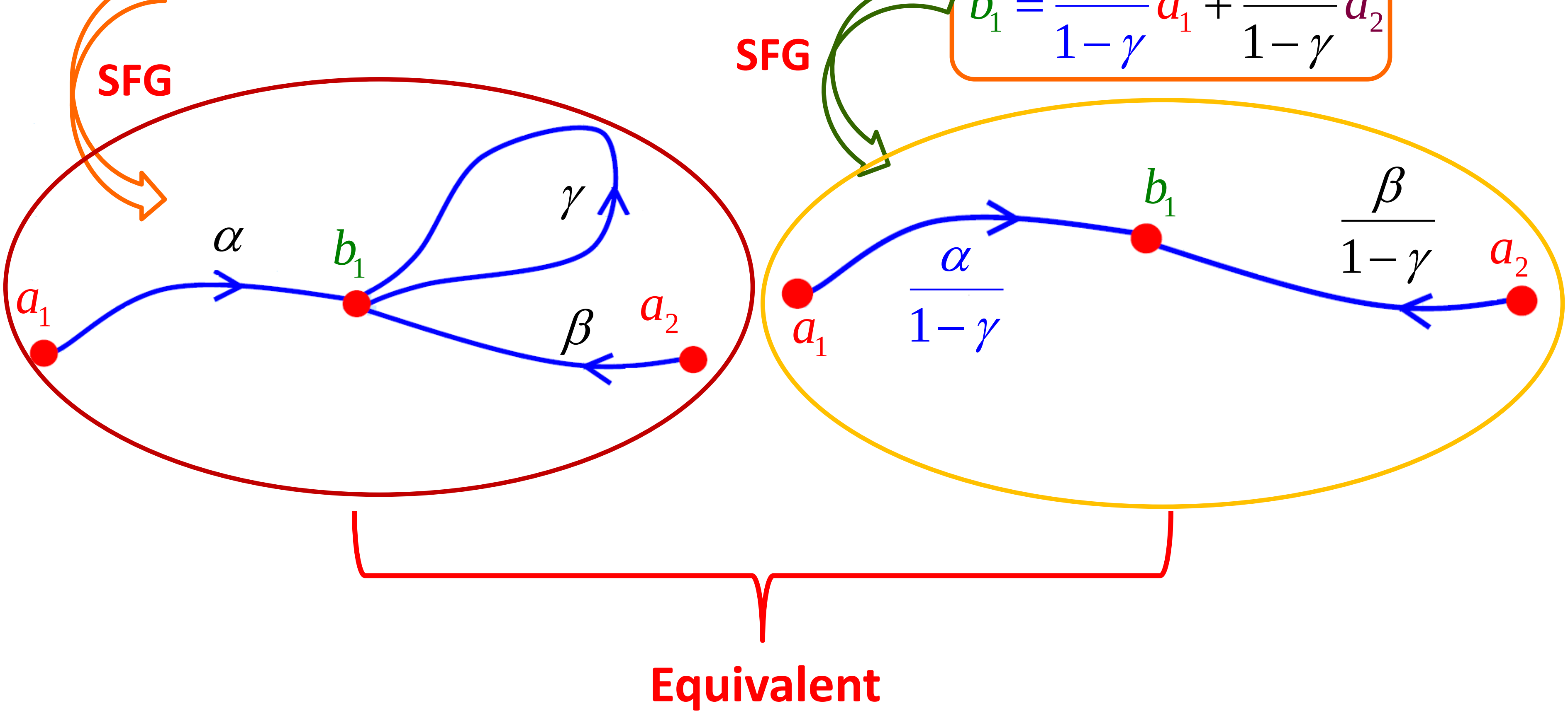
## Self-Loop Rule

- Consider the complex equation:

$$b_1 = \alpha a_1 + \beta a_2 + \gamma b_1$$

- A little bit of **algebra** allows us to determine the value of node $b_1$:

$$b_1 = \frac{\alpha}{1-\gamma} a_1 + \frac{\beta}{1-\gamma} a_2$$



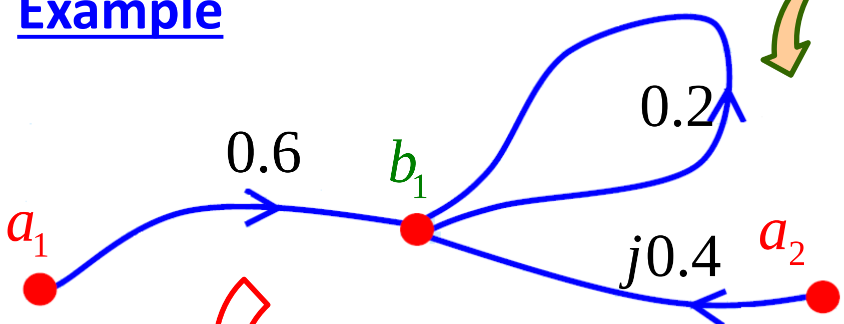
# Signal Flow Graph (contd.)

<u>This last discussion leads us to our **third** SFG **reduction rule:**</u>

**Rule 3 – Self-Loop Rule**

A self-loop can be eliminate by multiplying **all** of the branches "**feeding**" the self-loop node by $1(1-S_{sl})$, where $S_{sl}$ is the value of the self loop branch.

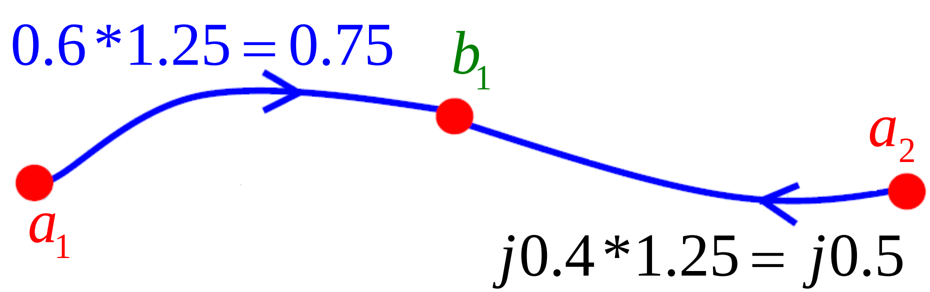
**<u>Example</u>**




can be simplified by **eliminating the self-loop** → multiply **both** of the two branches **feeding** the self-loop node by:

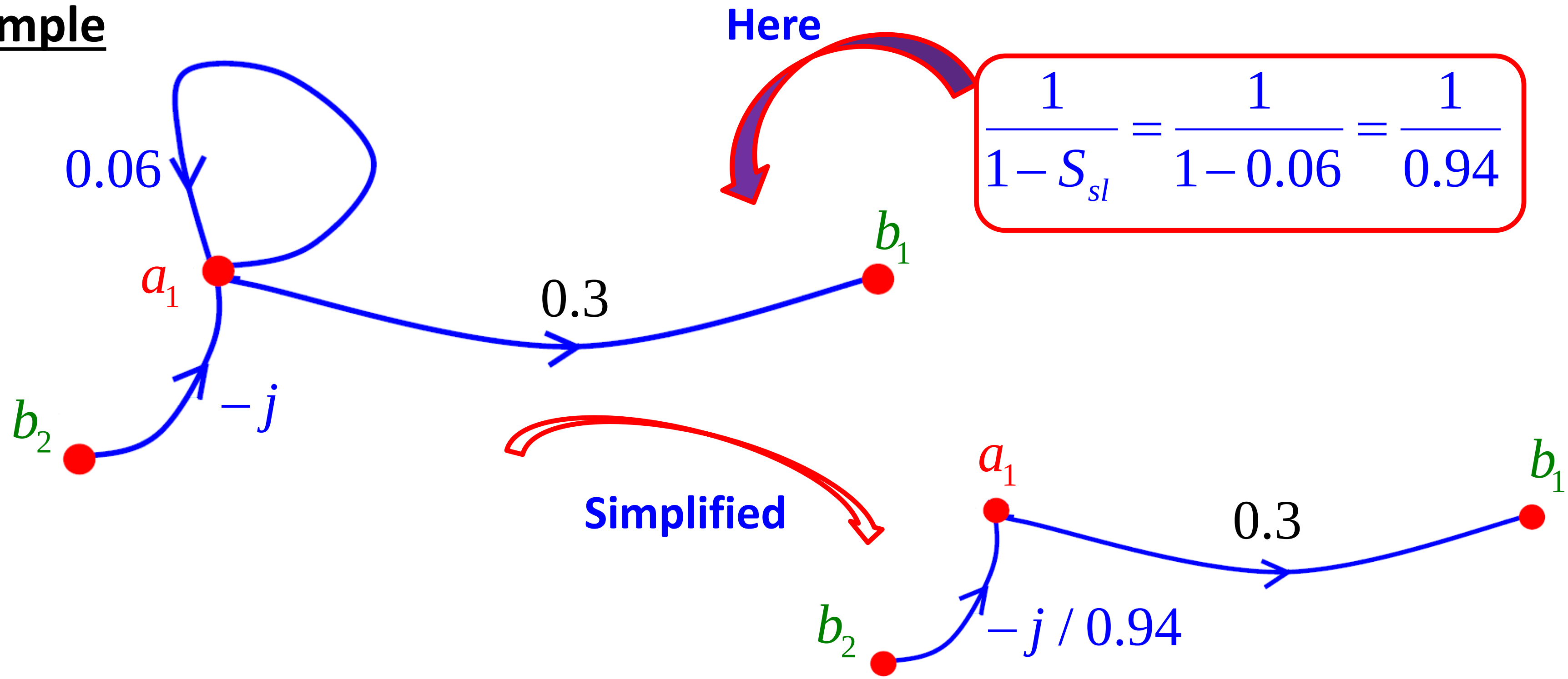
$$\frac{1}{1-S_{sl}}=\frac{1}{1-0.2}=1.25$$

**Simplified and Reduced**



# Signal Flow Graph (contd.)

**Example**




$$\frac{1}{1-S_{sl}}=\frac{1}{1-0.06}=\frac{1}{0.94}$$

Only the incoming branches are modified by the self-loop rule! Here, the 0.3 branch is **exiting** the self-loop node $a_1$ and therefore doesn't get modified. **Only** the –j branch(incoming at node $a_1$) to the self-loop node are modified by the self-loop rule!