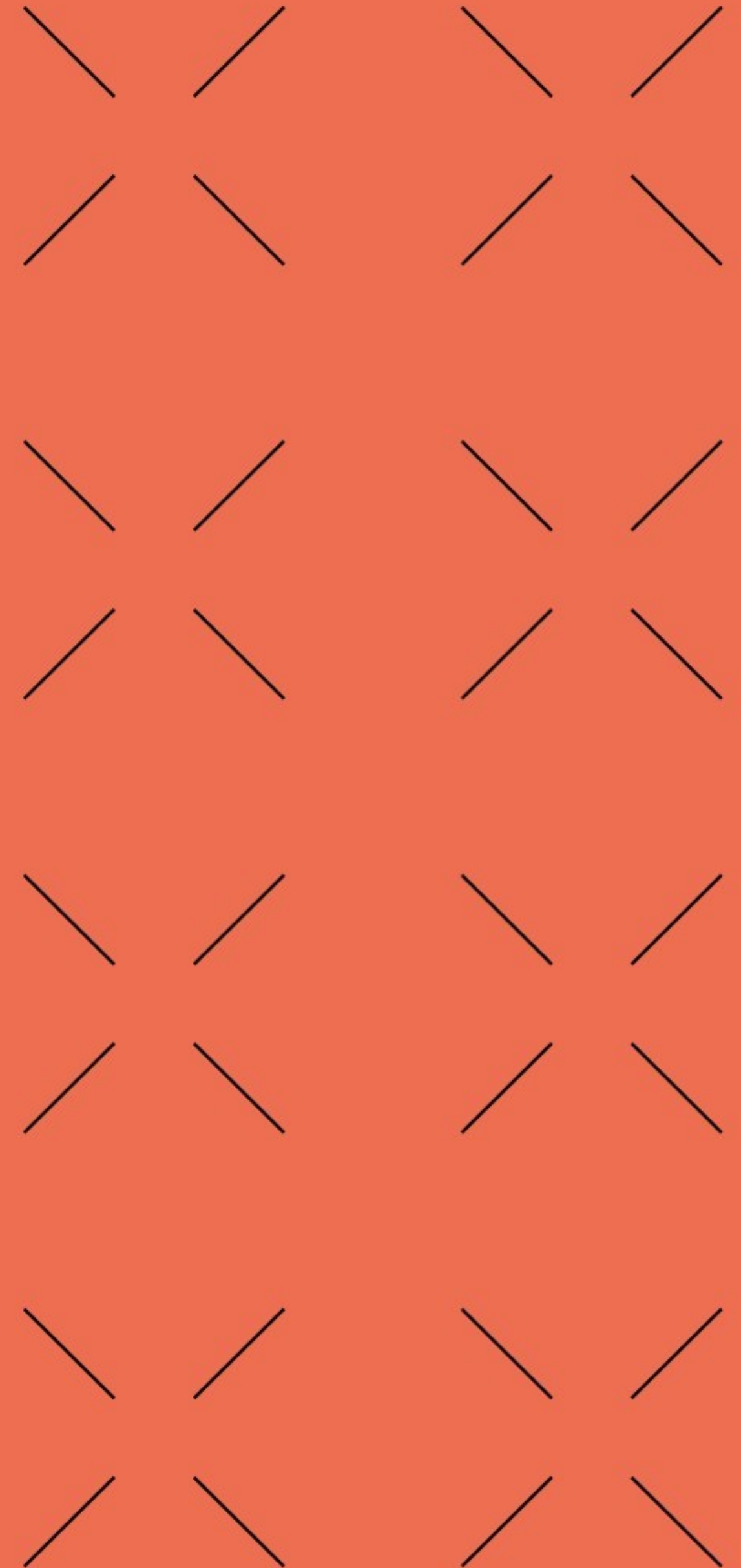


# Unidad 1.2

## Los procesos

Miguel Oliver – 2024 (Adriana Marti)



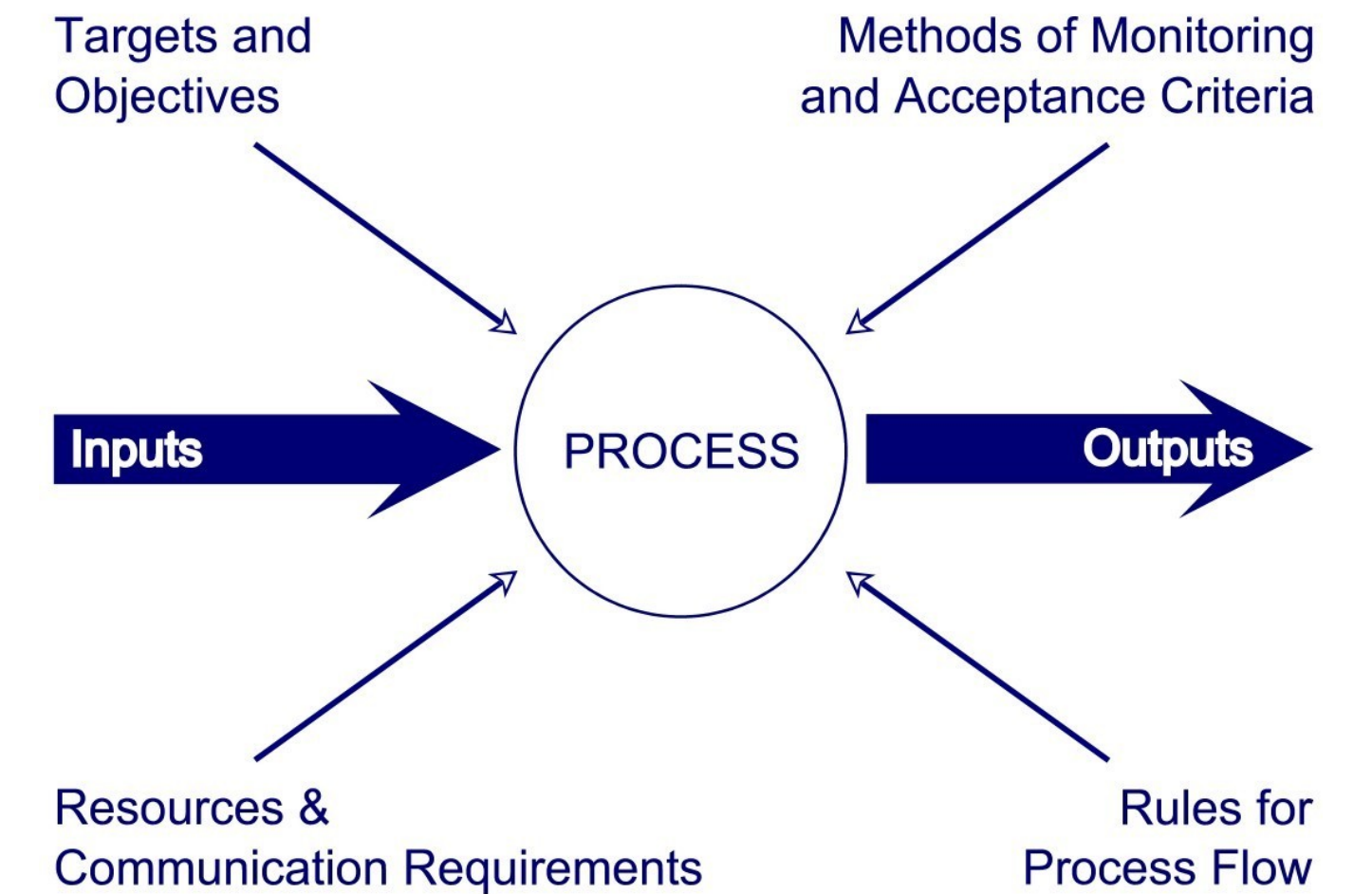
# Interacción de procesos

# Ejecutar programa



By Ecommdesign  
<https://iconscout.com/free-icon/buy-257>

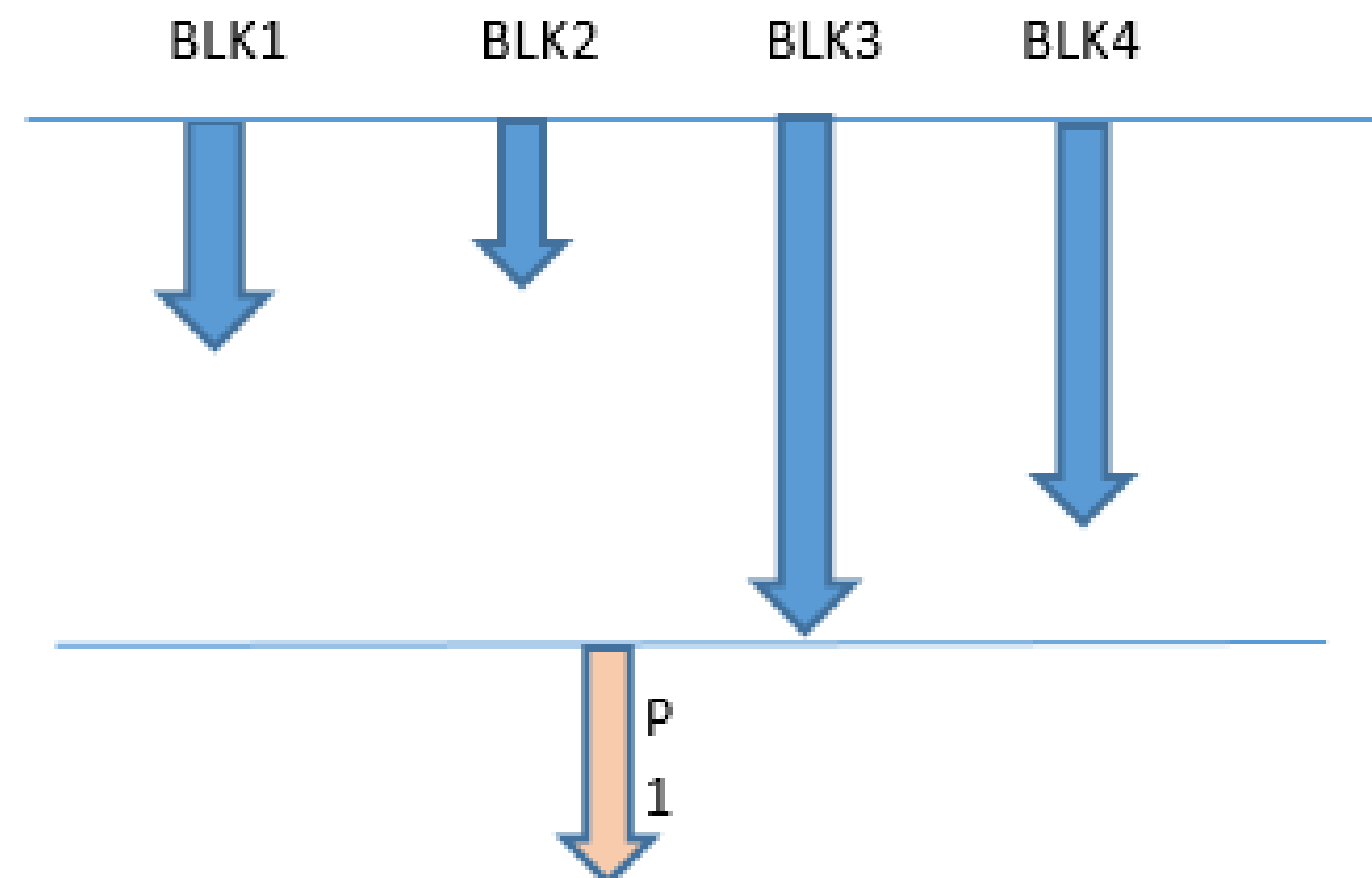
# Comunicación entre procesos



Andygalloway, CC BY-SA 4.0  
<<https://creativecommons.org/licenses/by-sa/4.0/>>, via  
Wikimedia Commons

# Esperar a que otros procesos terminen

Fork - Join



By Sini Balakrishnan  
<https://vlsi.pro/system-verilog-fork-join/>

# CONTROL DE ERRORES

# Control de errores



La explosión del cohete Arian 5:  
<https://www.youtube.com/watch?v=5tJPXYA0Nec>

Imagen: Spotting973, CC BY-SA 2.0 <<https://creativecommons.org/licenses/by-sa/2.0>>, via Wikimedia Commons

# MÉTODOS PARA EL CONTROL DE ERRORES

- **Try**

- try:**

- Código a ejecutar

- except** <tipo de error>:

- Se ejecutara si el código en try lanza un error

- else:**

- Se ejecutara si el código en try se ejecuta sin errores

- finally:**

- Esta parte se ejecutara siempre tanto si ha habido error como si no ha habido error

- **With**

Simplifica el control de errores en algunos casos (ficheros, sockets, bases de datos, entre otros). Te aseguras que los recursos son liberados en caso de error.

- with** Código a ejecutar **as** variable:

- Código a ejecutar



# MÉTODOS PARA EL CONTROL DE ERRORES

```
file = open('file_path', 'w')  
file.write('hello world !')
```

- **Try**

```
file = open('file_path', 'w')  
try:  
    file.write('hello world')  
finally:  
    file.close()
```

- **With**

```
with open('file_path', 'w') as file:  
    file.write('hello world !')
```

Si hay errores cierra el fichero file

# MÉTODOS PARA EL CONTROL DE ERRORES

## Combinar los dos métodos:

```
try:
    file = open('file_path', 'w')
except OSError as e:
    print("Error al abrir el fichero: ",e)
else:
    with file:
        file.write('hello world !')
```