# الف) تفاوت بین سیگنال‌ها و متغیرها در VHDL و کاربرد هرکدام

### سیگنال‌ها (Signals):

- تعریف: سیگنال‌ها عناصری هستند که بین فرآیندها و کامپوننت‌ها ارتباط برقرار می‌کنند.

- زمان بندی: تغییرات سیگنال‌ها با تاخیر انجام می‌شود (در پایان فرآیند یا بعد از تاخیر مشخص).

- محدوده: در (architecture) یا (package) تعریف می‌شوند.

- عملگر انتساب: از `<=` برای انتساب استفاده می‌شود.

- کاربرد:

- برای ارتباط بین فرآیندها

- پیاده‌سازی عناصر حافظه مانند فلیپ‌فلاپ‌ها

- مدل‌سازی خطوط ارتباطی در سخت‌افزار

### متغیرها (Variables):

- تعریف: متغیرها برای ذخیره‌سازی موقت داده‌ها در یک فرآیند استفاده می‌شوند.

- زمان بندی: تغییرات فوراً اعمال می‌شود (بلافاصله پس از انتساب).

- محدوده: فقط در داخل فرآیندها، توابع یا روال‌ها تعریف می‌شوند.

- عملگر انتساب: از `:=` برای انتساب استفاده می‌شود.

- کاربرد:

- برای محاسبات موقت در فرآیندها

- پیاده‌سازی الگوریتم‌های پیچیده

- شمارنده‌های حلقه‌ها

- ذخیره‌سازی نتایج میانی محاسبات

# ب) مقادیر ابتدایی و نهایی در 3 سیکل کلاک

### مقادیر اولیه (قبل از اولین سیکل کلاک):

- `signal\_a` = '0'

- `signal\_b` = '0'

- `var\_a` = '0'

- `var\_b` = '0'

سیکل کلاک 1:

1. `signal\_a` <= `var\_a` ('0')

2. `signal\_b` <= `var\_b` ('0')

3. `var\_a` := not `var\_a` ('1')

4. `signal\_b` := not `signal\_b` ('1') - این تغییر فقط روی نسخه محلی اثر دارد

### مقادیر پس از سیکل 1:

- `signal\_a` = '0' (تغییر در پایان سیکل اعمال می‌شود)

- `signal\_b` = '0' (تغییر با `<=` بر تغییر با `:=` غالب می‌شود)

- `var\_a` = '1'

- `var\_b` = '0'

سیکل کلاک 2:

1. `signal\_a` <= `var\_a` ('1')

2. `signal\_b` <= `var\_b` ('0')

3. `var\_a` := not `var\_a` ('0')

4. `signal\_b` := not `signal\_b` ('1') - باز هم فقط نسخه محلی

### مقادیر پس از سیکل 2:

- `signal\_a` = '1'

- `signal\_b` = '0'

- `var\_a` = '0'

- `var\_b` = '0'

سیکل کلاک 3:

1. `signal\_a` <= `var\_a` ('0')

2. `signal\_b` <= `var\_b` ('0')

3. `var\_a` := not `var\_a` ('1')

4. `signal\_b` := not `signal\_b` ('1') - نسخه محلی

### مقادیر پس از سیکل 3:

- `signal\_a` = '0'

- `signal\_b` = '0'

- `var\_a` = '1'

- `var\_b` = '0'