

# Prozess – Einsatzgebiete

---

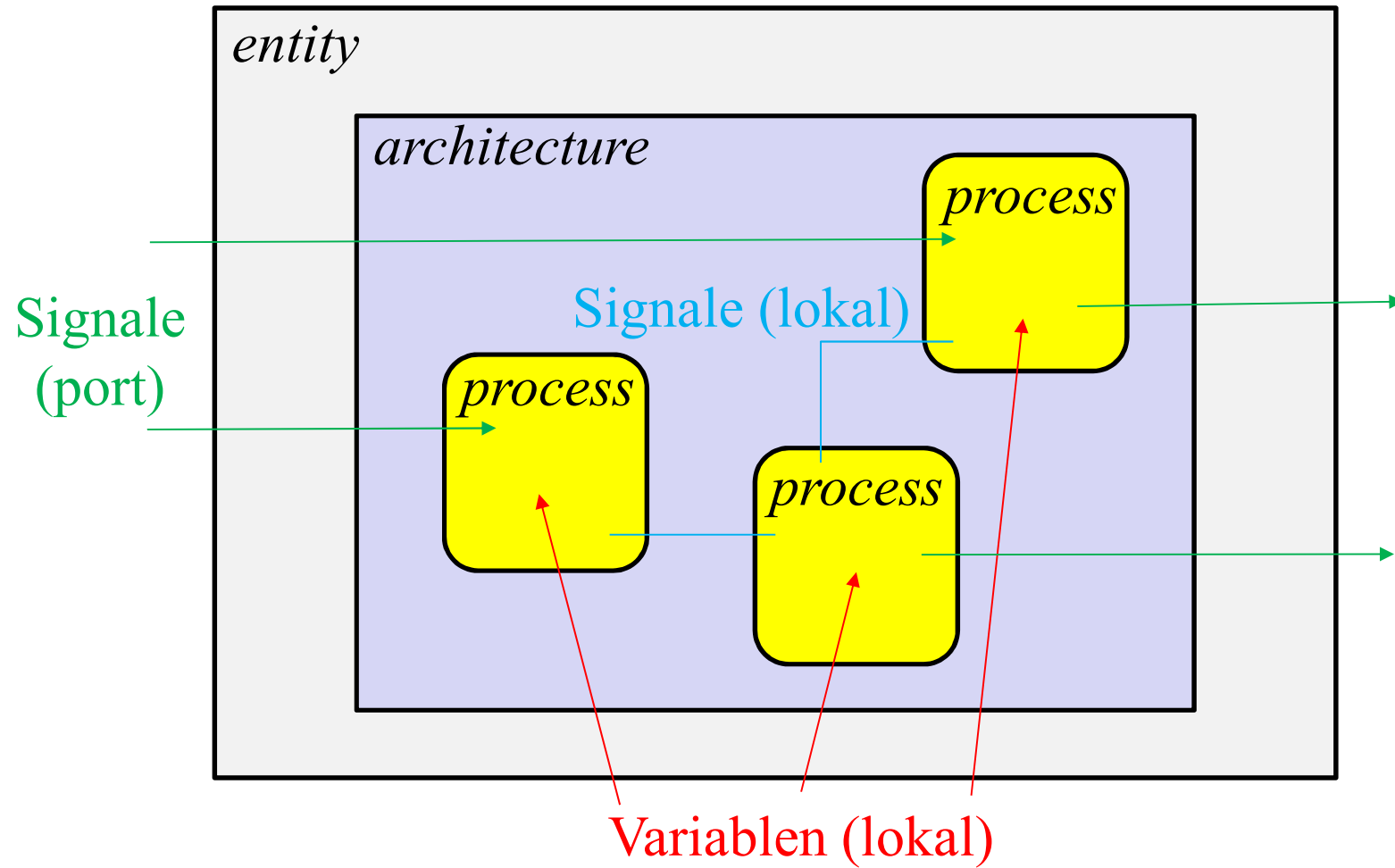
- **Getakteter Prozess**
  - Beschreibung eines Speichers
  - Anwendung: Automat
- **Kombinatorischer Prozess**
  - Nutzung „bequemerer“ Anweisungen
- **Nicht synthetisierbarer Prozess**
  - Erzeugung von Signalverläufen im Simulator
  - Anwendung: Testbench

# Prozess – Einordnung

---

- **Beinhaltet eine sequentielle Umgebung**
  - Andere Anweisungen (if/then, case/when)
  - Reihenfolge der Anweisungen hat Effekt
- **Ist nicht schachtelbar (kein Prozess im Prozess)**
- **Kann lokale Variablen deklarieren**
  - Formal wie Signale, aber andere Eigenschaften
- **Wird gleichzeitig mit anderen Prozessen ausgeführt**
  - Beliebig viele Prozesse in einer architecture möglich
- **Kommuniziert über Signale mit der Außenwelt**

# Prozess – Einordnung



# Prozess - Syntax

---

- Aufbau ähnlich einer *architecture*
- Kann lokale Deklarationen enthalten (keine neuen Signale)
- Kann eine Empfindlichkeitsliste (*liste*) enthalten  
(ab VHDL2008 mit (*all*) alle Signale möglich)
- Kann mit einem *label* versehen werden

```
label: process (liste) is
    -- lokale Deklarationen
begin
    -- sequentielle Anweisungen

end process label;
```

# if/then/else

---

```
if bedingung1
then
    anweisungen;
else
    if bedingung2
    then
        anweisungen;
    end if;
end if;
```

```
if bedingung1
then
    anweisungen;
elseif bedingung2
then
    anweisungen;
end if;
```

# case/when

---

**case** signal/variable **is**

**when** wert1 => Sequentielle  
Anweisungen

**when** wert2 => Sequentielle  
Anweisungen

**when** others => Sequentielle  
Anweisungen

**end case;**

Wie switch/case in C

Defaultfall **others** zwingend  
nötig

Kann geschachtelt werden,  
pro Fall Block von Anwei-  
sungen möglich

wert1, .., wertN müssen  
disjunkt sein

# Unterschiedliche Anweisungen

---

Nebenläufig	Sequentiell
Signale	Signale Variablen
Zuweisung <= (an Signal)	Zuweisung <= (an Signal) := (an Variable)
when/else	if/then/else, when/else ab VHDL 2008
with/select	case/when with/Select ab VHDL2008

# Signale vs. Variablen

---

Eigenschaft	Signal	Variable
Deklaration	entity, architecture	process, function
Syntax Zuweisung	S <= ...	V := ...
Zeitverlauf (im Simulator)	ja	nein
Zuweisung wird <b>im process wirksam</b>	<b>am Ende</b>	<b>sofort</b>



# Aufbau für synthetisierbare Schaltung

---

```
process (set, reset, clock)
```

```
begin
```

```
    if (bedingung aus set und reset)
    then
        -- asynchrone Aktionen
    elsif rising_edge(clock)
    then
        -- synchrone Aktionen
    end if;
```

```
end process;
```

# Hinweise

---

- **Taktabfrage**
  - **nur einmal** im process (siehe Template)
  - **nur eine Flanke** rising\_edge() oder falling\_edge()
  - **keine Verknüpfung des Taktsignals** mit anderen Signalen:  
clk\_i <= clk and hold; if rising\_edge(clk\_i) ...
  - Taktsignal sollte vom Typ std\_logic sein  
(ab VHDL2008 auch boolean)
- **Vorbelegung aller Variablen/Signale sinnvoll**
  - können später überschrieben werden (sequentielle Umgebung!)