



**Artgraphics**

# SystemVerilog基礎講座 タスクとファンクション

篠塚一也

アートグラフィックス

Document Revision: 1.0,2025.04.18

[www.artgraphics.co.jp](http://www.artgraphics.co.jp)

# 注意事項（Caveat）

---

- SystemVerilogの知識を個人的に習得する目的として本資料を活用して下さい。本資料を通して、業務（実践）で必要となるSystemVerilogに関する知識を習得して頂くのが本来の目的です。
- 転用目的（本来の目的と違った他の用途に使う事）で本資料を使用する事はご遠慮下さい。また、本資料から学んだ知識を転載する場合等は出典が本資料である事を明記して下さい。但し、他の著者の文書にも書かれている内容は、この限りではありません。本注意事項は現在及び過去に於ける弊社からの全てのフリーダウンロード資料に適用されます。
- 本注意事項に合意出来ない場合には、本資料を速やかに抹消して下さい。尚、ダウンロード記録は、依然として残ります。

# 例題ー1 タスクとファンクションに関する問題

---

- 以下のようにタスクとファンクションの定義が与えられたとします。ただし、それぞれの内容は省略されています。以下の問に答えて下さい。

---

```
task create_process(process proc);  
...  
endtask  
  
function check(x,y);  
...  
endfunction
```

---

- ① タスクcreate\_processのポートprocの方向は何ですか？
- ② ファンクションcheckの戻り値の型とビット幅は何ですか？
- ③ ファンクションcheckのポートxとyのデータタイプは何ですか？

# 例題の解答と解説

---

- ① タスクcreate\_processのポートprocの方向はinputです。
- ② ファンクションcheckの戻り値の型はlogicで1ビットです。
- ③ ファンクションcheckのポートxとyのデータタイプは、いずれもlogic型です。

## 例題ー2 タスクの定義に関する問題

---

- 以下に示すように、forkで二つのプロセスを生成していますが、タスクprocの定義法は正しいですか？正しくなければ訂正して下さい。

---

```
module test;

  initial begin
    fork
      proc("proc1");
      proc("proc2");
    join_none
  end

  task proc(string msg);
  ...
endtask

endmodule
```

---

# 例題の解答と解説

---

- タスクprocは、二つの異なるプロセスから同時に呼び出されています。したがって、procはリエントラントの構造を持たなければなりません。しかし、procはスタテックなタスクとして定義されているので間違いです。
- 以下のように、automaticを指定する必要があります。

---

```
module test;

    initial begin
        fork
            proc("proc1");
            proc("proc2");
        join_none
    end

    task automatic proc(string msg);
    ...
    endtask

endmodule
```

---

## 例題ー3 ファンクションに関する問題

---

- 以下に示すファンクションは、1からnまでの整数の和を求めています。このファンクションは正しく動作しますか？

---

```
module test;
...
function int sum(int unsigned n);
    if( n < 1 )
        sum = 0;
    else
        sum = sum(n-1)+n;
endfunction
...
endmodule
```

---

# 例題の解答と解説

---

- このファンクションは正しく動作しません。
- 再起呼び出しをしているので、このファンクションはリエントラントでなければなりません。したがって、以下のようにautomaticを指定しなければなりません。

---

```
function automatic int sum(int unsigned n);  
    if( n < 1 )  
        sum = 0;  
    else  
        sum = sum(n-1)+n;  
endfunction
```

---



# 参考文献

---

文献[1]は最新版の仕様書です。是非一読下さい。SystemVerilogに関する知識の確認には、文献[2-4]を参照して下さい。文献[4]は設計分野で必要とされるSystemVerilogの基礎知識を非常に詳しく解説しています。

- [1] IEEE Std 1800-2023: IEEE Standard for SystemVerilog – Unified Hardware Design, Specification and Verification Language.
- [2] 篠塚一也、SystemVerilogによる検証の基礎、森北出版 2020.
- [3] 篠塚一也、SystemVerilog入門, 共立出版 2020.
- [4] 篠塚一也、SystemVerilog超入門, 共立出版 2023.