18. ASIC Design Guidelines

Introduction

• The following design guidelines have been adapted from [2]: European Silicon Structures (ES2), Zone Industrielle, 13106 France. Solo 2030 User Guide, e02a02 edition, June 1992

• These recommendations are useful in order to avoid functional faults and get the desired functionality
Synchronous Circuits (1)

- All data storage elements are clocked
- The same active edge of a single clock is applied at precisely the same time to all storage elements

Synchronous Circuits (2)

- **NON-RECOMMENDED CIRCUITS:**
  - Flip-flop driving clock input of another Flip-flop:
    - The clock-input of the second FF is skewed by the clock-to-q delay of the first FF and not activated at every activation clock edge (e.g. ripple counter)
Synchronous Circuits (3)

• NON-RECOMMENDED CIRCUITS:
  – Gated clock line:

  ![Gated Clock Line Diagram]

  – Clock skew caused by gating the clock line (e.g. multiplexer in clock line)

Synchronous Circuits (4)

• NON-RECOMMENDED CIRCUITS:
  – Double-edged clocking:

  ![Double-Edged Clocking Diagram]

  – FFs are clocked on the opposite edges of the clock signal
  – Insertion of scan-path impossible
  – Difficulties in determining critical path lengths
Synchronous Circuits (5)

- **NON-RECOMMENDED CIRCUITS:**
  - Flip-flop driving asynchronous reset of another Flip-flop:
    - Synchronous design principle, that all FFs change state at exactly the same time is not fulfilled

**Recommended Circuits** will be described during the following sections

Clock Buffering (1)

- **NON-RECOMMENDED CIRCUITS:**
  - Unequal depth of clock buffering:
    - causes clock skew
Clock Buffering (2)

• NON-RECOMMENDED CIRCUITS:
  – Unbalanced fanout of clock buffers:
  – Clock skew by different load-dependent delays
  – Excessive clock fanout should be avoided (slow edges)

Clock Buffering (3)

• Recommended circuits:
  – Balanced clock tree buffering
  – Same depth of buffering
  – Same fanout
  – Limited fanout in order to achieve sharp clock edges
Clock Buffering (4)

- **Recommended circuits:**
  - Combined geometric/tree buffering
  - Using intermediate buffer of suitable strength at each fanout point

Gated Clocks (1)

- **NON-RECOMMENDED CIRCUITS:**
  - Multiplexer on clock line:
    - Signal change at multiplexer input can cause a glitch at the clk input (FF captures invalid data)
    - Gating the clock line introduces clock skew
Gated Clocks (2)

- **Recommended circuits:**
  1) Enabled (E-type) flip-flop: 
  2) Toggle (T-type) flip-flop:

![Gated Clocks Diagram](image)

Double-edged Clocking (1)

- **NON-RECOMMENDED CIRCUITS:**
  - Pipelined logic with double-edged clocking:

![Double-edged Clocking Diagram](image)

  - Not recommended in context with scan-path methods
Double-edged Clocking (2)

- **Recommended circuits:**
  - Pipelined logic with single-edged clocking:

- **NON-RECOMMENDED CIRCUITS:**
  - Flip-flop driving the asynchronous reset of another flip-flop:
Asynchronous Resets (2)

- **Recommended circuits:**
  - Global asynchronous reset by external signal:

Asynchronous Resets (3)

- **Recommended circuits:**
  - Flip-flop driving the synchronous reset of another flip-flop:
Shift Registers (1)

• **NON-RECOMMENDED CIRCUITS:**
  - Shift register with forward or reverse chain of clock buffers:
    - Internal clock skew can cause data fallthrough

Shift Registers (2)

• **Recommended circuits:**
  - Shift register with balanced tree of clock buffers:
Asynchronous Inputs (1)

• NON-RECOMMENDED CIRCUITS:
  – Circuits with complicated feedback loops to capture asynchronous inputs (very sensitive to noise, and functionality can be influenced by placement and routing delays)

Asynchronous Inputs (2)

• Recommended circuits:
  – Chain of two or more D-type flip-flops for capturing an asynchronous input:

  \[\text{d (external)} \quad \text{d} \quad \text{d (internal)}\]

  – The probability of propagating a metastable state is decreased with increasing number of register stages
Asynchronous Inputs (3)

- **Recommended circuits:**
  - Use of 4-bit register as shift register for capturing an asynchronous input:

    ![Diagram of 4-bit register as shift register]

    - The probability of propagating a metastable state is decreased with increasing number of register stages

Asynchronous Inputs (4)

- **Recommended circuits:**
  - Asynchronous handshake circuit:

    ![Diagram of asynchronous handshake circuit]
Asynchronous Inputs (5)

- The asynchronous handshake circuit works as follows:
  a) The first flip-flop is reset asynchronously when the r input is zero or when the qb outputs of the second and the third FF both have the value 0
  b) The q-output of the first FF is asynchronously set to high, when a positive edge arises at its ck-input
  c) The high output of the first FF is propagated through the second and the third FF in the two following cycles. The q-outputs of these FFs are set to zero and the reset logic for the first FF is activated. Now the first FF is ready to receive another edge at its input.
  d) ...

Asynchronous Inputs (6)

d) Three cases of metastability caused by simultaneously rising edges of the asynchronous input and the system clock:
  1) the second FF stabilizes to q=1 before the next rising clock edge (circuit works as desired)
  2) the second FF settles to q=0 and the third FF remains in its state. Since the output q of the first FF is high, the propagation of this output works correctly, but it needs one cycle more than in the first case.
  3) The metastable state of the second FF is still there at the next rising edge of the clock signal. Then the third FF also becomes metastable. The probability of receiving a metastable d (internal) signal can be reduced by increasing the length of the register chain.
Asynchronous Inputs (7)

- Operation of asynchronous handshake circuit:

Delay Lines and Monostables (1)

- NON-RECOMMENDED CIRCUITS:
  - In general, it cannot be recommended to build circuits with a functionality that relies on delays.
  - E.g. monostable pulse generator:
Delay Lines and Monostables (2)

- **NON-RECOMMENDED CIRCUITS:**
  - Pulse generator using flip-flop:
    
    ![Pulse generator using flip-flop](image)

  - Multivibrator:
    
    ![Multivibrator](image)

Delay Lines and Monostables (3)

- **Recommended circuits:**
  - Synchronous pulse generator:
    
    ![Synchronous pulse generator](image)

  - Usage of higher clock speed
  - Minimum time resolution is given by clock cycle
Bistable Elements (1)

- **NON-RECOMMENDED CIRCUITS:**
  - Cross-coupled flip-flops and RS-flip-flops
  - Bistable storing elements formed by cross-coupled NAND or NOR gates:

Bistable Elements (2)

- **NON-RECOMMENDED CIRCUITS:**
  - Asynchronous RS-flip-flop:
Bistable Elements (3)

- **Recommended circuits:**
  - Use D-types with set/reset
  - Use latch configured as RS flip-flop:

\[
\begin{array}{c}
\text{(active low)} \\
\text{s} \\
\text{0} \\
\text{d} \\
\text{q} \\
\text{q} \\
\text{r} \\
\text{ld qb} \\
\text{qb} \\
\text{(active high)}
\end{array}
\]

RAMs and ROMs in Synchronous Circuits 1

- **Problem:** RAMs are double-edge triggered. The address is latched on the opposite edge to the data
- **Timing scheme:**

\[
\begin{array}{c}
\text{ME} \\
\text{(RAM/DPRAM)} \\
\text{latch address} \\
\text{latch data} \\
\text{addr setup} \\
\text{addr hold} \\
\text{data setup} \\
\text{data hold} \\
\text{WEbar (write)} \\
\text{WEbar (read)}
\end{array}
\]
**RAMs and ROMs in Synchronous Circuits 2**

- **Recommended circuits:**
  - Interfacing RAM into synchronous circuit: ME and WEbar generation

![Circuit Diagram]

**RAMs and ROMs in Synchronous Circuits 3**

- **Recommended circuits:**
  - Using flip-flop for WEbar generation: timing scheme

![Timing Diagram]
**RAMs and ROMs in Synchronous Circuits 4**

- **Recommended circuits:**
  - Avoiding floating RAM/DPRAM output propagation

![Recommended circuit diagram](image)

**Tristates (1)**

- **NON-RECOMMENDED CIRCUITS:**
  - Tristate bus with non-central enable control:
Tristates (2)

- **Recommended circuits:**
  - Tristate bus with central control of all tristate enable signals and one additional driver that is activated on non-controlled states

Tristates vs. Multiplexer

<table>
<thead>
<tr>
<th>Tristates:</th>
<th>Multiplexer:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- large area</td>
<td>- small area</td>
</tr>
<tr>
<td>- limited buffering</td>
<td>- efficient routing</td>
</tr>
<tr>
<td>- large routing load → slow</td>
<td></td>
</tr>
</tbody>
</table>

- Control decoding expense is the same for tristates and multiplexers.
- → Multiplexers are more favourable
Parallel Signals

- **NON-RECOMMENDED CIRCUITS:**
  - Wired-OR part used to create higher fanout:

```
+--------+          +--------+
| x      |          | y      |
+--------+          +--------+
   inv3   |          |      |
   inv3   |
```

- **Recommended Circuits:**
  - High-fanout buffer replacing wired OR part

```
+--------+          +--------+
| x      |          | y      |
+--------+          +--------+
              |
      inv6   |
```

Fanout (1)

- **NON-RECOMMENDED CIRCUITS:**
  - Excessive fanout on control signals:
**Fanout (2)**

- **Recommended circuits:**
  - Geometric buffering on control signal:

**Fanout (3)**

- **Recommended circuits:**
  - Tree buffering on control signal:
Design for Speed (1)

• Use a maximum of 2 inputs on all combinational logic gates:

  ![AND4 gate](image1.png)
  ![Equivalent faster NAND/NOR logic using 2-input gates](image2.png)

• Use AOI logic (complex cells from standard cell library) where possible. The figure below shows a multiplexer using AOI logic:

  ![Multiplexer using AOI logic](image3.png)

Design for Speed (2)

• Feed late changing inputs late into combinational logic:

  ![Late-changing inputs](image4.png)

• Use shift (Johnson) counters instead of binary counters:

  ![Shift (Johnson) counters](image5.png)

<table>
<thead>
<tr>
<th>q0</th>
<th>q1</th>
<th>q2</th>
<th>q3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Design for Speed (3)

• Use duplicate logic to reduce fanout:

- Non-optimised circuit:
  - Low fanout critical path
  - High fanout non-critical path

- Circuit optimised for speed:
  - Low fanout critical path
  - High fanout non-critical path

• Use fast library cells where available
• Reduce length of critical signal paths
• Use Schmitt trigger inputs in noisy environments

Design for Testability (1)

• Testability = Controllability + Observability

• NON-RECOMMENDED CIRCUITS:
  - Circuit with inaccessible internal logic: only the first block is controllable, and only the last block is directly observable
Design for Testability (2)

• **Recommended circuit:**
  – Insert test inputs and outputs

\[\text{Diagram of recommended circuit}\]

Design for Testability (3)

• **NON-RECOMMENDED CIRCUITS:**
  – Chain of counters: first counter is not directly observable and second counter is not directly controllable

\[\text{Diagram of non-recommended circuit}\]
Design for Testability (4)

• **Recommended circuit:**
  – Break long counter / shift register chains
  – Chain of counters broken by test input $tc$ and output signals:

![Diagram](image1)

Design for Testability (5)

• **NON-RECOMMENDED CIRCUITS:**
  – Counter with closed feedback loop: initial state is not known

![Diagram](image2)
Design for Testability (6)

- **Recommended circuit:**
  - Open feedback loops
  - Counter with feedback loop opened by test control \( tr \) and output signals:

![Diagram showing recommended circuit](image)

Design for Testability (7)

- **Recommended circuits:**
  - Use BIST (Built-In-Self-Test) with compiled megacells
  - Compiled megacell with compiled inputs/outputs:

![Diagram showing recommended circuit](image)
Design for Testability (8)

• Recommended circuits:
  – Scan path testing
  – E-type scan path flip-flop (right):
  – Circuit with scan path (below):

Design for Testability (9)

• Recommended circuits:
  – Use of JTAG boundary scan path
  – JTAG test circuitry: