Delay-Locked Loop
- Design Examples, Design Issues/Tips
Agenda

- **Introduction**
- Building blocks of DLL
- Design issues of DLL
- DLL examples
- Design examples
  - DLL for DDR SDRAM
  - DLL for direct RDRAM
- **Summary**
Why DLL in DRAM?

- tAC, tF and tS should be minimum.

<table>
<thead>
<tr>
<th>Delay to be compensated</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>tAC</td>
<td>Clock buffer &amp; data output buffer</td>
</tr>
<tr>
<td>tS</td>
<td>Data input buffer &amp; clock buffer</td>
</tr>
</tbody>
</table>

→ PLL & DLL can do this job.
Is DLL better than PLL for DRAM?

• Yes !!!
  – Frequency multiplication is not necessary.
  – There is no jitter accumulation in DLL.
  – DLL is a first-order system.
    - Inherently stable
    - Parasitic pole due to the feedback delay exists and thus the stability should be considered as well.
PLL & DLL Loop Topologies

- **PLL**
  - Second-order system
  - Phase alignment by VCO

- **DLL**
  - First-order system
  - Phase alignment by VCDL
Transient Response to Supply Step

- Jitter of VCO accumulates until the loop feedback’s correcting action takes effect.
- VCDL in DLL; no jitter accumulation
Delay Locked Loop

Ext. Clock

Variable Delay Line

Compensation Delay

PD

LPF

Ext. Data

DOUT Buffer

Int. Clock

Data from Array

d1

tCC - (d1 + d2)
d1 + d2

d2
Agenda

• Introduction
• Building blocks of DLL
• Design issues of DLL
• DLL examples
• Design examples
  – DLL for DDR SDRAM
  – DLL for direct RDRAM
• Summary
Variable Delay Line Schemes - 1

- Variable unit delay
  - Conventional

- Variable number of delay stages
  - Digital DLL (SMD, HPLD ...)

- Variable number of delay stages & variable unit delay
  - SEC DDR SDRAM
Variable Delay Line Schemes - 2

• Variable unit delay
  – Simple control.
  – Limited locking range due to stuck/harmonic-lock problem.

• Variable number of delay stages
  – Fast locking and/or fast standby-mode exit.
  – For wide locking range, large silicon area is required.
  – Inherently large skew (resolution = fixed unit delay).

• Variable number of delay stages & variable unit delay
  – Wide locking range.
  – Complex control.
Compensation Delay

• Compensation delay
  – Sum of the delays of input clock buffer, data buffer, and internal clock buffer.

• Loading and bias differences between original buffers and replica buffers.
  – Compensation delay cannot be exactly the same as the original delay.
  – Fuse/metal options for post-tuning.
Wafer Level Comp. Delay Control

Hitachi, SOVC00
Delay Cell

- Single-ended delay cell
  - Simple
  - Dynamic power only (no static current)

- Differential delay cell
  - Complex biasing
  - Static power consumption
  - Immune to supply noise and thus smaller jitter

- Variables for delay control
  - Current
  - Capacitance
  - Resistance
  - Voltage swing
Single-Ended Delay Cell

Variable loading capacitance

Pull-down strength control
Differential Delay Cell

- Replica biasing ensures constant voltage swing independent of control voltage.
Phase Detector - D F/F

- D F/F can be used as a phase detector.
- Bang-bang jitter.
- Uncertainty window as large as set-up/hold window of D F/F.
Phase Detector - PFD

A

B

Up

Dn

Reset

D

Q

Up=0

Dn=1

Up=0

Dn=0

Up=1

Dn=0

A

B

A

B

Up

Dn

Reset

D

Q

A

B

Up

Dn

A

B

Up

Dn

Chang sik Yoo @ DRAM Design 4, Samsung Electronics
Direct Phase Comparison

- External clock is directly compared with the replica clock.
- No need for replica input clock buffer.
- Slew rate independent skew control.

Hitachi, SOVC00
Charge Pump

- In analog DLL, PD output is converted to charge by charge pump.
- Charge pump output is low-pass filtered by a capacitor.
Loop Filter

• In conventional analog DLL, a capacitor is sufficient.
• For fast standby-mode exit, locking information can be stored in a digital code.
  – In digitally controlled DLL, a digital low-pass filter is necessary.
  – Feedback delay should be considered when designing the digital low-pass filter.
Agenda

• Introduction
• Building blocks of DLL
• **Design issues of DLL**
• DLL examples
• Design examples
  – DLL for DDR SDRAM
  – DLL for direct RDRAM
• Summary
Delay Range Problem of DLL - 1

- DLL cannot distinguish between $tD = \Delta$ & $tD = tCC + \Delta$.

$$tD = \text{delay from Ref-CLK to DLL-CLK}$$
$$tCC = \text{clock period}$$
$$tVCDL_{max} = \text{max. delay of VCDL}$$
$$tVCDL_{min} = \text{min. delay of VCDL}$$
Delay Range Problem of DLL - 2

• To prevent stuck/harmonic-lock problem, the following relationship should be ensured.

\[
0.5 \times t_{CC} < t_{VCDL\_min} < t_{CC}, \ t_{CC} < t_{VCDL\_max} < 1.5 \times t_{CC}
\]

\[
\rightarrow \ \text{Max} \ ( t_{VCDL\_min}, \ 2/3 \times t_{VCDL\_max} ) < t_{CC}
\]

\[
< \text{Min} \ ( 2 \times t_{VCDL\_min}, \ t_{VCDL\_max} )
\]

\[
\rightarrow \ \text{Limited locking range.}
\]
Stuck/Harmonic-Lock Free DLL - 1

- For wide locking range, control range of tVCDL should be as large as possible.

Wide tVCDL range leads to stuck/harmonic-lock problem.
Stuck/Harmonic-Lock Free DLL - 2

- Initial locking starts with $t_{VCDL\_\text{max}}$ or $t_{VCDL\_\text{min}}$.
  - If locking starts from $t_{VCDL\_\text{min}}$ or $t_{VCDL\_\text{max}}$, it is clear that $t_{VCDL}$ should be increased/decreased till DLL is locked.
- Use a PD which can prevent stuck/harmonic-lock problem.
- Develop a DLL which can find the stuck/harmonic-lock free initial condition.
Stuck/Harmonic-Lock Free PD

- With conventional PD, case (a) and (b) would give the same PD output, and thus stuck/harmonic-lock occurs.
Stuck/Harmonic-Lock Free DLL - Ex.

- tRDC is settled at 1/8 x tCC by replica delay line.
  → Stuck/harmonic-free locking range is greatly increased.
Locking Time

- **Closed-loop DLL**
  - Locking time > several hundred clock cycles
  - Conventional DLL with feedback loop
- **Open-loop DLL**
  - Locking time < several clock cycles
  - Synchronous mirror delay (SMD) type
Fast-Lock Technique - Ex. 1

- Fast-lock by successive approximation < 64 cycles
- Counter-mode operation during normal cycle
Fast-Lock Technique - Ex. 1 (cont’d)

Successive approximation mode \(\leftrightarrow\) Counter mode

Ext-CLK

CLK2

PHASE

slow fast slow slow

DCLK2

CBn

CB7 set CB6 set CB0 set \(\pm\) LSB

Int-CLK

MSB LSB

8 cycles * 8 bits = 64 cycles
Fast-Lock Technique - Ex. 2

Patent Pending, SEC

Changsik Yoo @ DRAM Design 4, Samsung Electronics
DLL Locking Time for DDR SDRAM

- In DDR SDRAM, DLL should be locked within 200 cycles after power-up sequence.
  - Not a hard job.
  - In DDR SDRAM, DLL is necessary only during read cycle.
  - For small standby current, it is desirable to turn off DLL except read cycle which requires DLL to be able to provide stable clock within 4 cycles after turn-on.
  - Initial locking right after power-up need not be faster than 200 cycles.
  - Fast standby-mode exit is the key issue.
Agenda

• Introduction
• Building blocks of DLL
• Design issues of DLL
• DLL examples
• Design examples
  – DLL for DDR SDRAM
  – DLL for direct RDRAM
• Summary
Synchronous Mirror Delay (SMD) - 1

Open-loop system; fast locking < 2 cycles
Area penalty for wide freq. range and locking accuracy.
Synchronous Mirror Delay (SMD) - 2

- Ext. Clock
- Input Buffer
- Meas. Delay Line
- Var. Delay Line
- Int. Clock

\[ t_{CC} - (d1 + d2) \]

\[ 2 * t_{CC} - (d1 + d2) \]
Hierarchical Phase-Locking Delay

Number of Delay Stage @ 50MHz (tCC=20ns)

<table>
<thead>
<tr>
<th>Type</th>
<th>Formula</th>
<th>Stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventional (Delay ELEM. 0.2ns)</td>
<td>20ns / 0.2ns = 100 (stage)</td>
<td></td>
</tr>
<tr>
<td>Hierarchical (Coarse Delay ELEM.: 1ns, Fine Delay ELEM.: 0.2ns)</td>
<td>20ns / 1ns + 1ns / 0.2ns = 25 (stage)</td>
<td></td>
</tr>
</tbody>
</table>
Bi-Directional Delay (BDD)

😊 Open-loop system;
→ Fast locking < 2 cycles
😊 High resolution
😄 Area penalty
Delay step size is too large.
Register Controlled DLL - 2

Fujitsu, ISSCC97,98

😊 Locking information is stored as a digital code.
   ➔ Fast standby-mode exit
😊 High resolution thanks to vernier type delay line.

Changsik Yoo @ DRAM Design 4, Samsung Electronics
## Loop Configurations of DLL

<table>
<thead>
<tr>
<th></th>
<th>Initial lock</th>
<th>Active lock</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>180°</td>
<td>Coarse</td>
</tr>
<tr>
<td>64M DDR</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>128M DDR</td>
<td>Variable</td>
<td>no. of stages</td>
</tr>
<tr>
<td>A</td>
<td>Variable</td>
<td>no. of stages</td>
</tr>
<tr>
<td>B</td>
<td>Variable</td>
<td>no. of stages</td>
</tr>
<tr>
<td>C</td>
<td>Variable</td>
<td>no. of stages</td>
</tr>
<tr>
<td>D</td>
<td>Variable</td>
<td>no. of stages</td>
</tr>
<tr>
<td>RAMBUS</td>
<td>Unit delay + Selection (Reference loop)</td>
<td>interpolation</td>
</tr>
<tr>
<td>E</td>
<td>Unit delay + Selection (Reference loop)</td>
<td>unit delay</td>
</tr>
</tbody>
</table>
Agenda

• Introduction
• Building blocks of DLL
• Design issues of DLL
• DLL examples

• Design examples
  – DLL for DDR SDRAM
  – DLL for direct RDRAM

• Summary
DLL for DDR SDRAM

- **Wide locking range**
  - tCC : 6ns ~ 15ns

- **Jitter specification is not tough.**
  - tAC, tDQSCCK : +/-0.75ns
  - tDQSQ : +/-0.5ns @ tCC = 7.5ns
  - But, most of the jitter comes from SSO noise of DOUT buffer and thus the DLL jitter should be minimized.

- **Locking time < 200 cycles**

- **Power consumption < 20mA**
  - As small as possible
Hybrid DLL for DDR SDRAM - 1

SEC, ISSCC'99
Hybrid DLL for DDR SDRAM - 2

- Coarse locking + Fine locking = Wide freq. range

Coarse locking finds the optimum number of delay cells for the specific tCC.

Fine locking controls the skew to be < several 10ps.
Hybrid DLL for DDR SDRAM - 3

\[ \Delta t_{\text{CLK}} = t_{\text{CLK1}} - t_{\text{CLK2}} \text{ (ns)} \]

Digital \hspace{1cm} \text{Analog}

Jitter \sim 20 \text{ps}

CLK1 \hspace{1cm} \Delta t_{\text{CLK}} \hspace{1cm} CLK2

Time (\mu s)
Most of the jitter comes from SSO noise of VDDQ/VSSQ. (not from DLL)
DLL for Direct RAMBUS DRAM - 1

- Direct RDRAM channel structure
  - Source synchronous data transmission
    - Tx/Rx on both edges of clock
    - Tx is in quadrature

<table>
<thead>
<tr>
<th></th>
<th>Rx clock</th>
<th>Tx clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller</td>
<td>CTM</td>
<td>CFM</td>
</tr>
<tr>
<td>RDRAM</td>
<td>CFM</td>
<td>CTM</td>
</tr>
</tbody>
</table>

Diagram showing clock and data waveforms with labels.
• DLL purpose
  – On-chip buffer delay compensation.
  – Quadrature phase generation for transmission clock.

Phase relationship between CTM and CFM varies with channel location.
DLL Operation in RDRAM

- Reference loop
  - Multi-phase clock generation.
  - Locking information is stored as a digital code which is converted to analog quantity by a D/A converter.
  - Reference loop is shared by Rx/Tx fine loops.

- Rx/Tx fine loop
  - Locking information is stored as a digital code which controls the phase mixer.

→ Major locking information is stored as a digital code and thus fast standby-mode exit.
Rx Clock Generation in RDRAM

CTM is used as the clock source because it is closer to the external clock source than CFM.
Tx Clock Generation in RDRAM
90° Phase Generation
Phase Mixing

- Similar to variable delay element.
- Mixes two 45° spaced signals.
- Mixing weight is controlled by $I_{K1}$ and $I_{K2}$.
Duty Cycle Correction

- For maximum valid data window
- DCC information is stored in a loop capacitor.

- If tHigh = tLow, control voltage is stabilized.
Agenda

• Introduction
• Building blocks of DLL
• Design issues of DLL
• DLL examples
  – DLL for DDR SDRAM
  – DLL for direct RDRAM
• Summary
Summary

- DLL should be designed considering the system requirement.
  - Locking time
  - Power consumption
  - Standby-mode exit
  - Area