Transistor Level Implementation of Data Weighted Averaging (DWA)

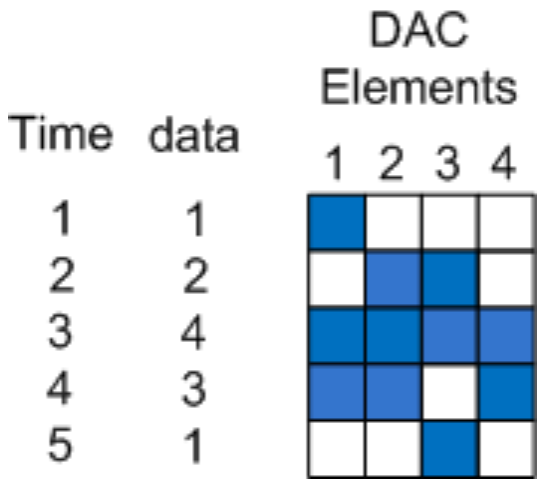
Weilun Shen

05/28/2010

A ΔΣ modulator with multi-bit quantizer

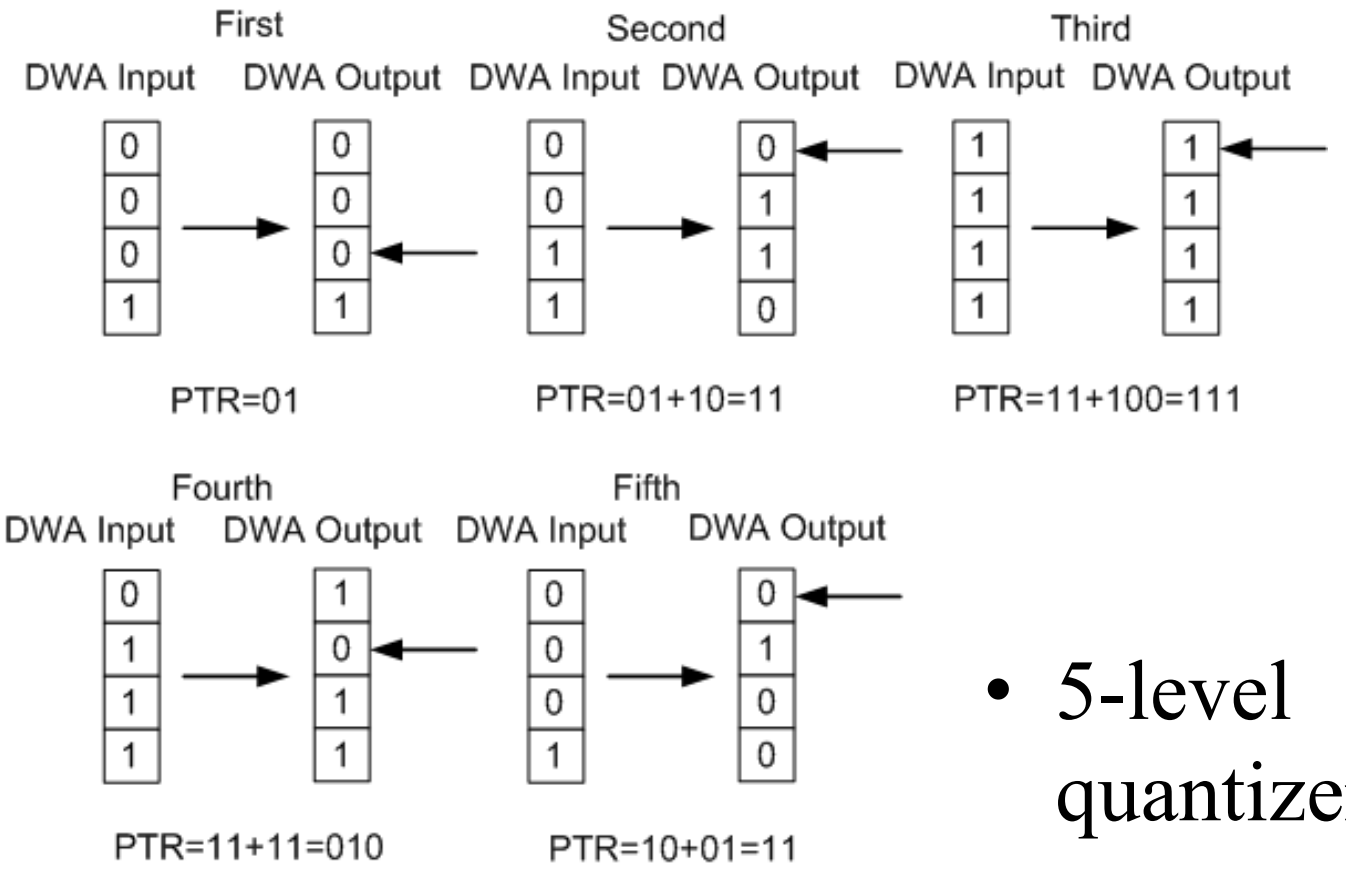
- To improve the stability of the modulator
- To improve the SQNR under low OSR
- DEM to linearize the DAC at the modulator input
- Data weighted averaging (DWA) is a very effective and simple way to linearize the feedback DAC

The DWA algorithm



- Rotation of the thermometer code
- First order noise shaping for DAC mismatch

DWA implementation



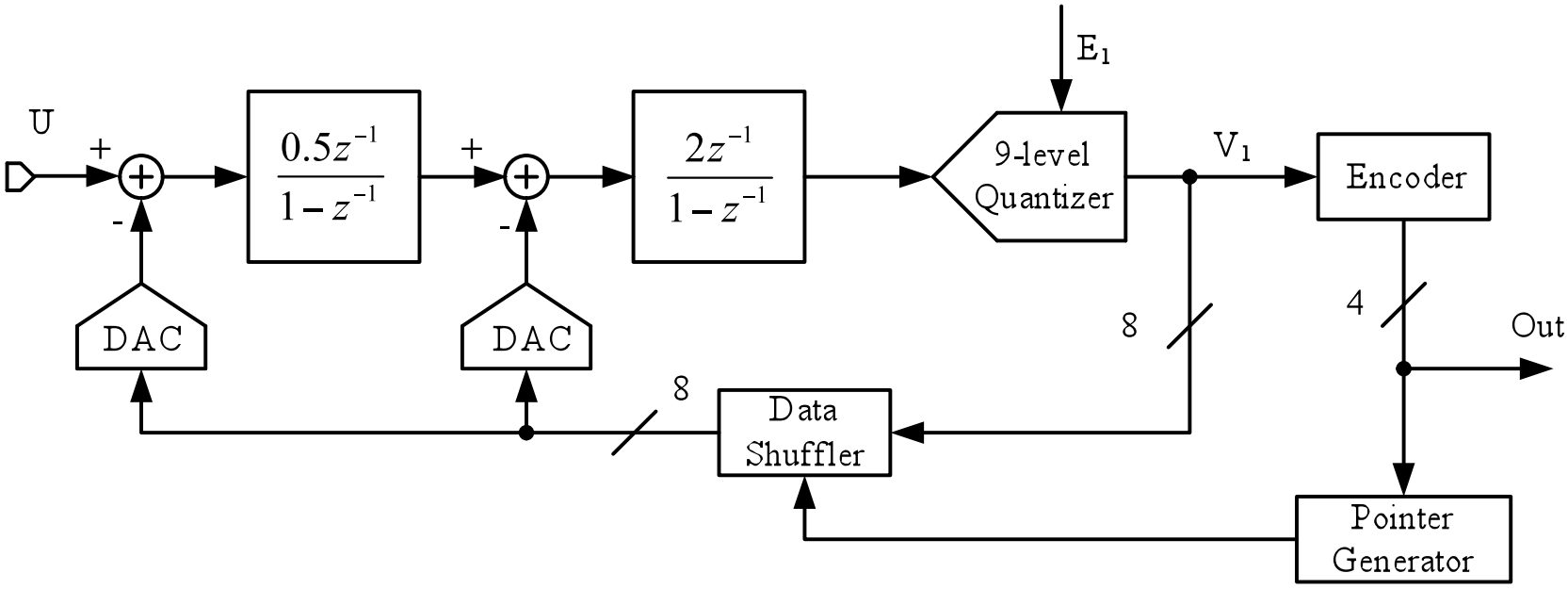
- 5-level quantizer as an example

DWA implementation

- A DWA system contains two important building blocks
- The first one is the pointer which indicates which unit element shall be used as the starting point in a DAC operation
- The second one is a data shuffler which maps the relationship between the thermometer code and DAC unit elements

ΔΣ modulator with multi-bit quantizer

- A well designed DWA only includes a shuffler in the feedback path while the pointer calculation is outside the modulator loop [1]



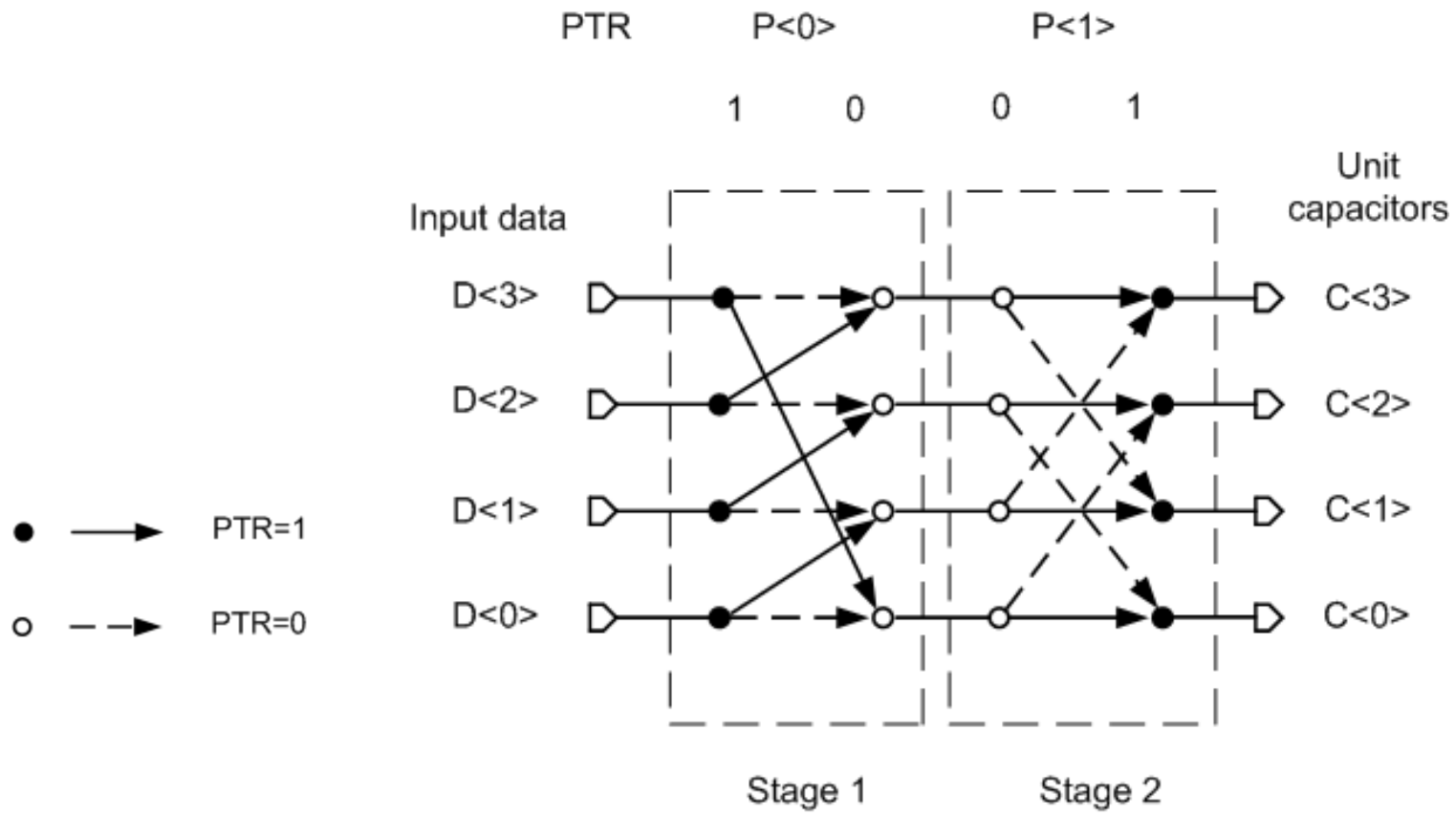
DWA pointer generation

- Needs binary code, so there is a thermometer-to-binary code converter;
- Add all previous quantizer outputs in the binary code domain, and store them as the pointer for the next DWA operation;
- To perform these, an accumulator and a register are needed

Data shuffler

- Two kinds of data shufflers are commonly used:
- Logarithmic shifter
- Barrel shifter.

Logarithmic shifter

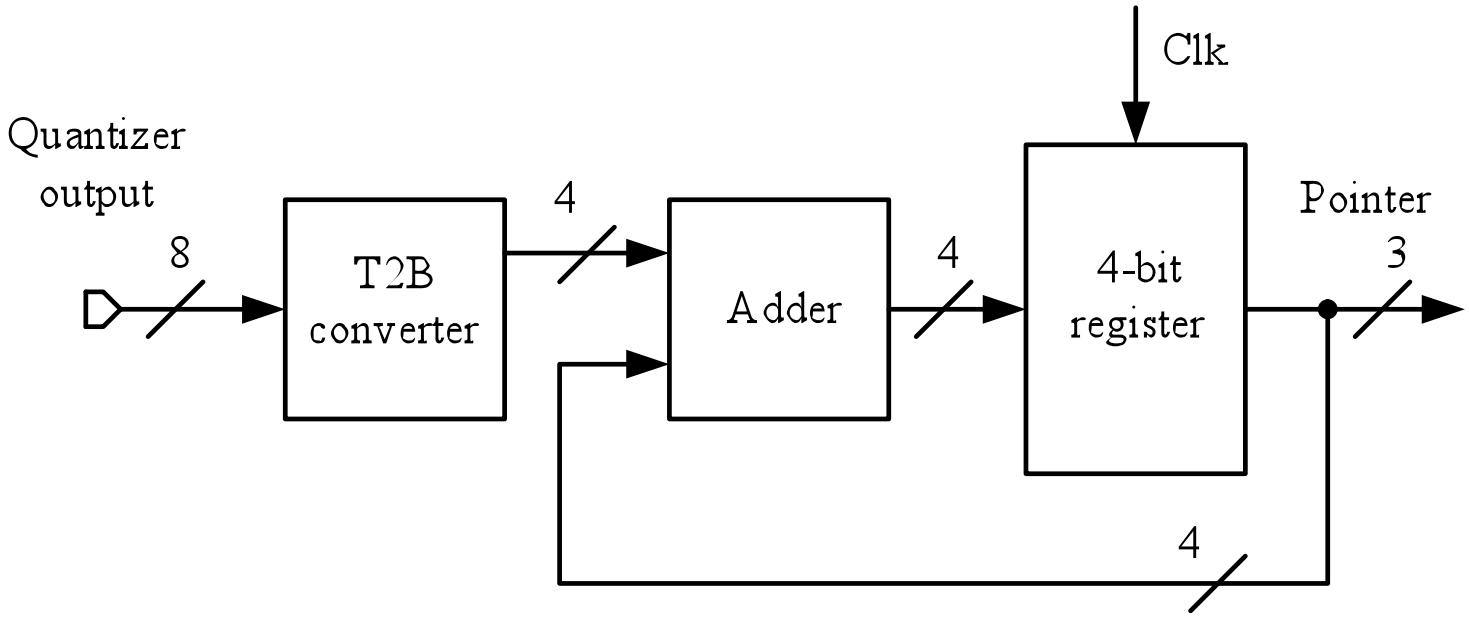


Logarithmic shifter

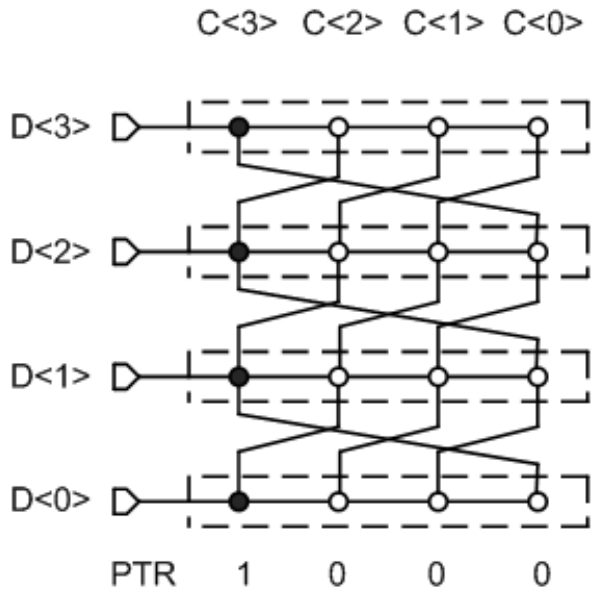
- Multi-stage design
- Speed is only moderate, but the total number of switches can be kept fairly small
- Total number of switches: $2*(N-1)*\log_2(N-1)$, where N is the number of quantizer levels
- Pointer is directly generated usinn an accumulator.

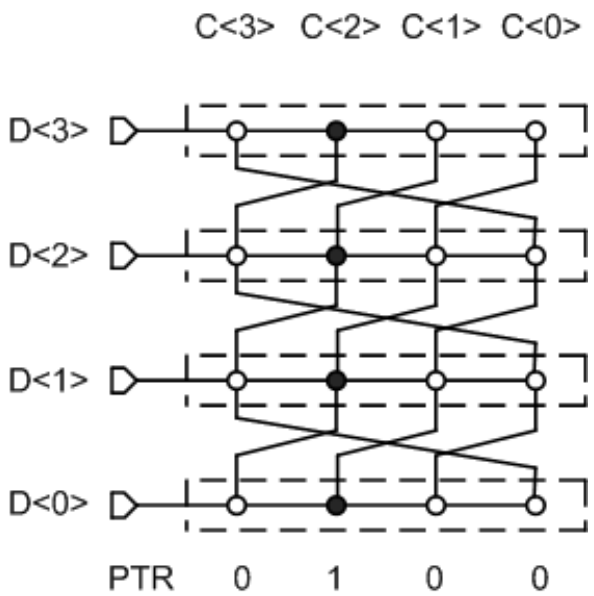
DWA pointer generation

- Pointer generator for a logarithmic shifter (9 level quantizer)



Barrel shifter





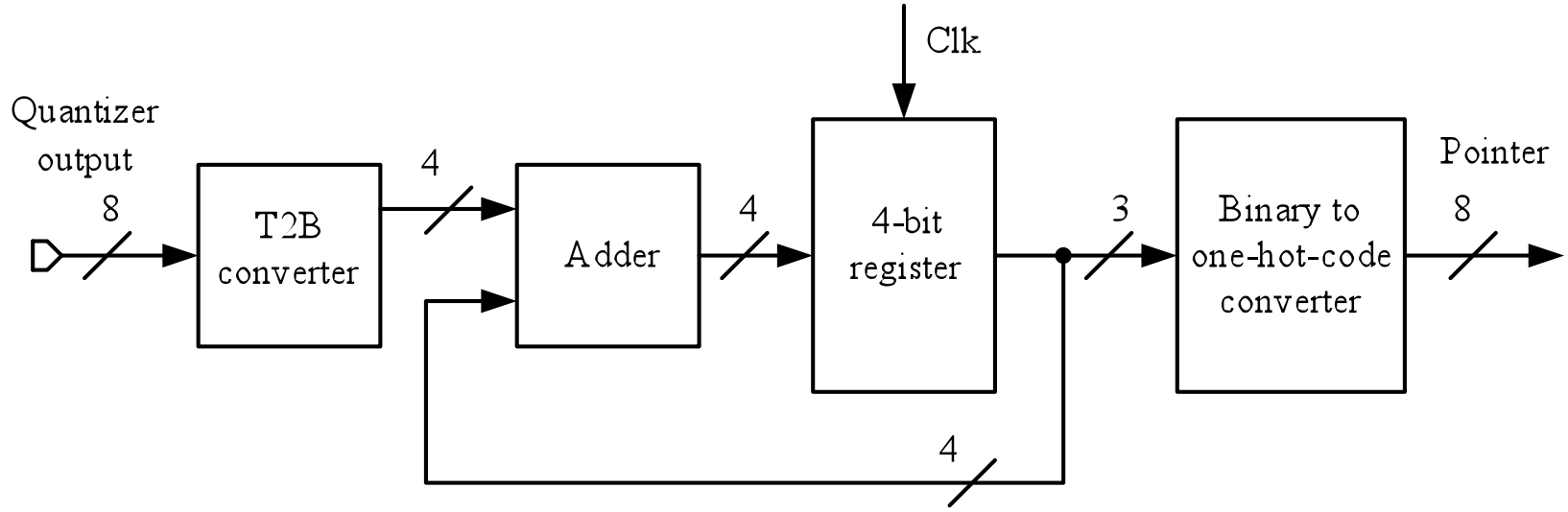
PTR	1000	0100	0010	0001
C<3>	D<3>	D<0>	D<1>	D<2>
C<2>	D<2>	D<3>	D<0>	D<1>
C<1>	D<1>	D<2>	D<3>	D<0>
C<0>	D<0>	D<1>	D<2>	D<3>

Barrel shifter

- Single-stage design, so it is faster;
- The number of switches $(N-1)^2$ grows rapidly with higher number of bits in the DAC;
- Pointer needs to be decoded.

DWA pointer generation

- Pointer generator for a barrel shifter (9 level quantizer)



Some design issues

- Switches can be implemented with transmission gates because it is able to pass both logic “0” and logic “1”
- Switch RC time constant can be calculated as

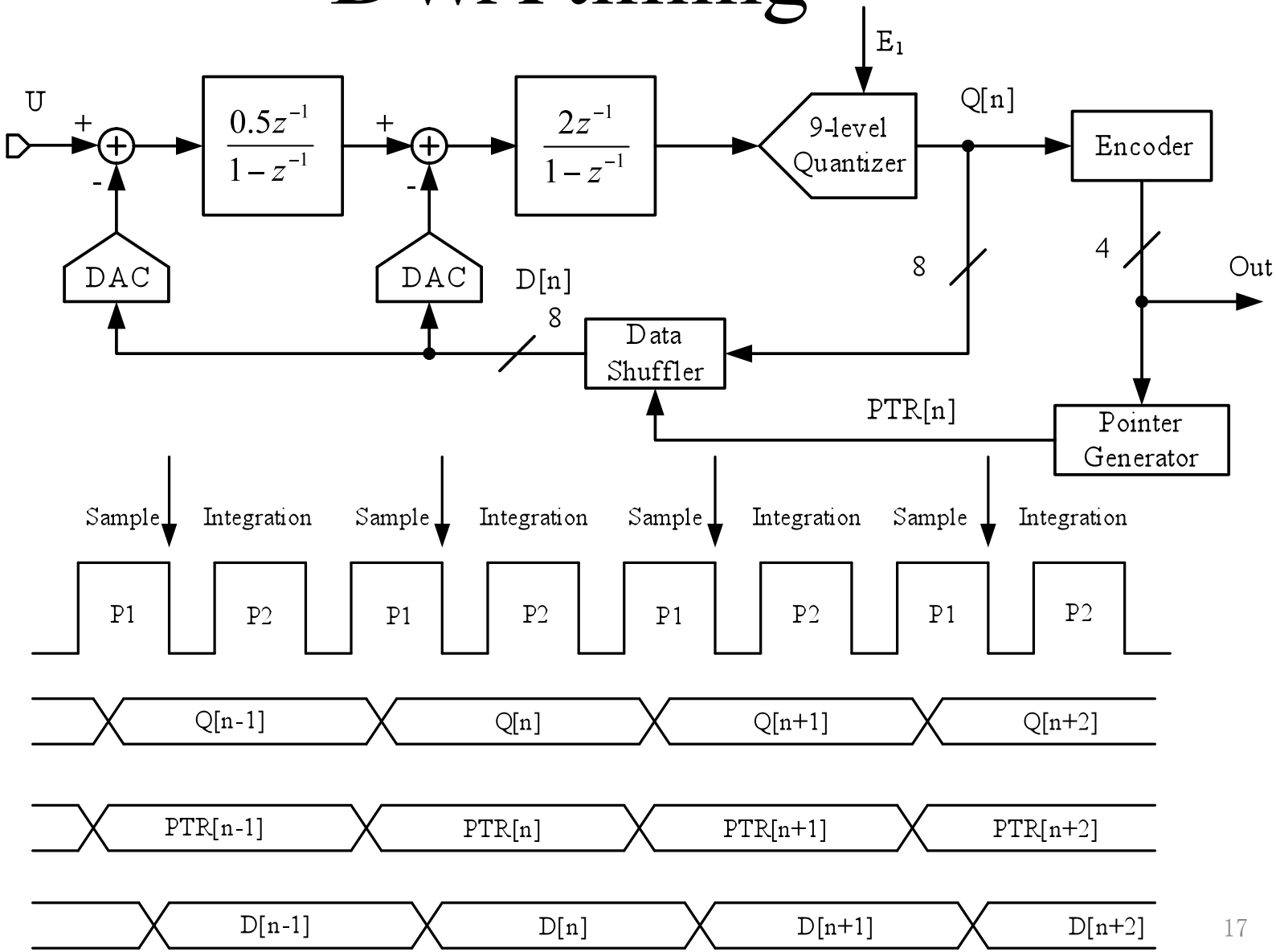
$$\tau = \frac{1}{\mu C_{ox} \frac{W}{L}(V_{DD} - V_{TH})} \cdot C_{ox} \cdot WL = \frac{L^2}{\mu(V_{DD} - V_{TH})}$$

- This time constant only depends on mobility, power supply, threshold voltage and channel length. All of them are heavily technology related

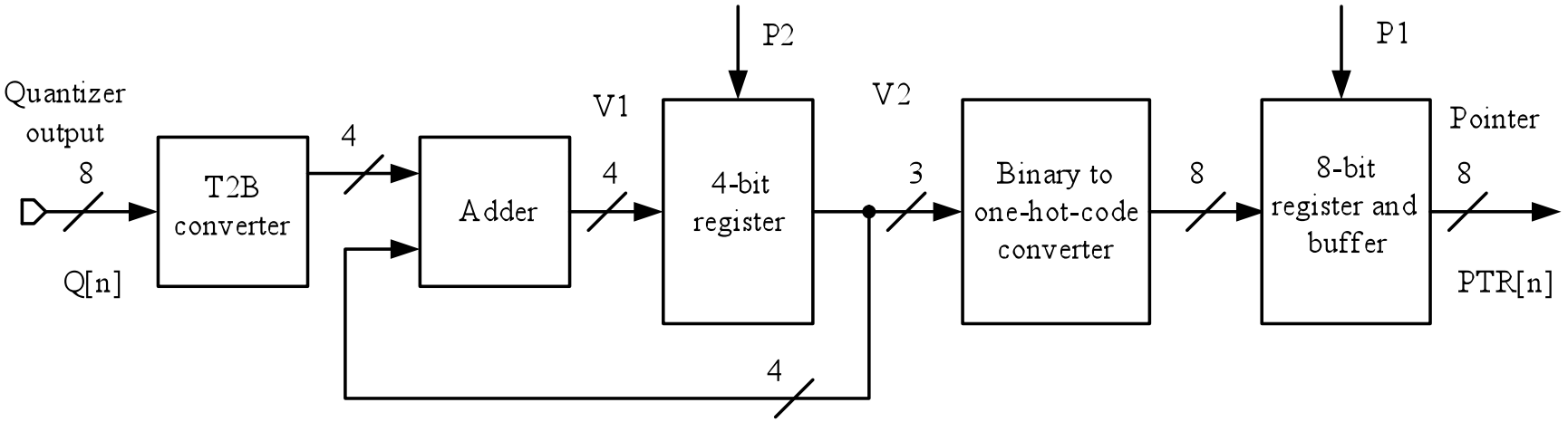
Some design issues

- Use minimum channel length to improve speed
- Switch RC time constant decreases with advanced CMOS technology
- Switch channel width should not be too small. Very small transistors in advanced technology such as 90nm or 65nm show little intrinsic parasitic.
- However, with small transistors, after layout is done, wiring parasitic will significantly lower the RC time constant.
- Switch aspect ratio W/L can be 6-10, according to my own design experiences.

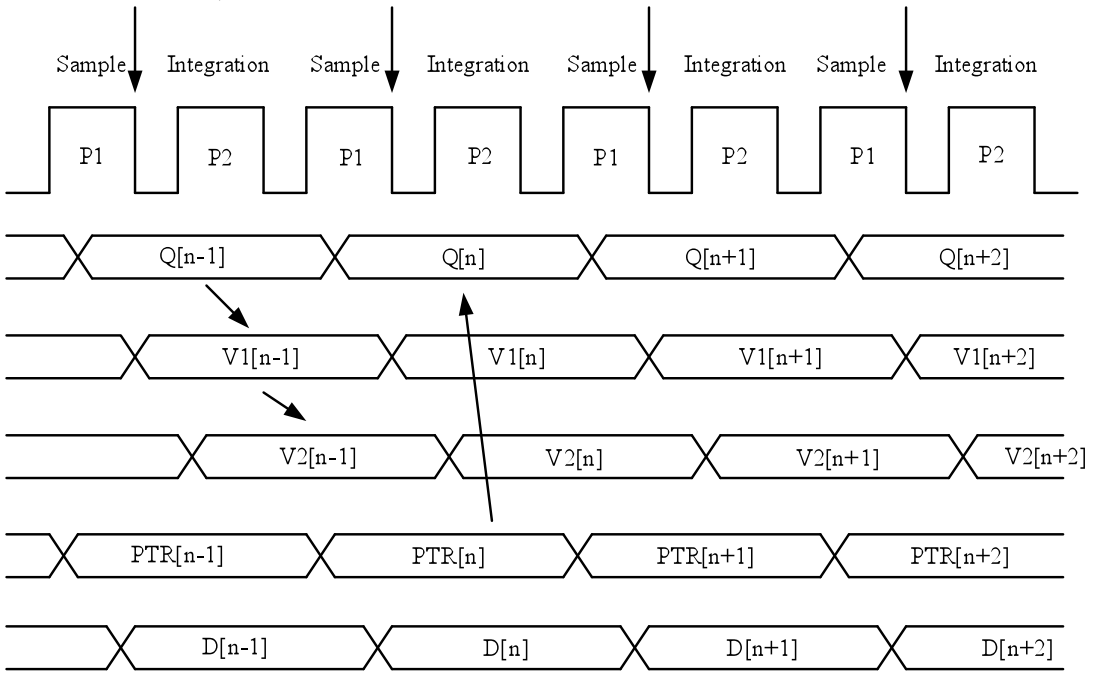
DWA timing


U
+
-
$\frac{0.5z^{-1}}{1-z^{-1}}$
+
-
$\frac{2z^{-1}}{1-z^{-1}}$
E_1
9-level Quantizer
Q[n]
Encoder
4
Out
8
DAC
DAC
D[n]
8
Data Shuffler
PTR[n]
Pointer Generator
Sample
Integration
Sample
Integration
Sample
Integration
Sample
Integration
P1
P2
P1
P2
P1
P2
P1
P2
Q[n-1]
Q[n]
Q[n+1]
Q[n+2]
PTR[n-1]
PTR[n]
PTR[n+1]
PTR[n+2]
D[n-1]
D[n]
D[n+1]
D[n+2]

DWA timing



- DWA pointer generator for a 9-level quantizer with barrel shifter

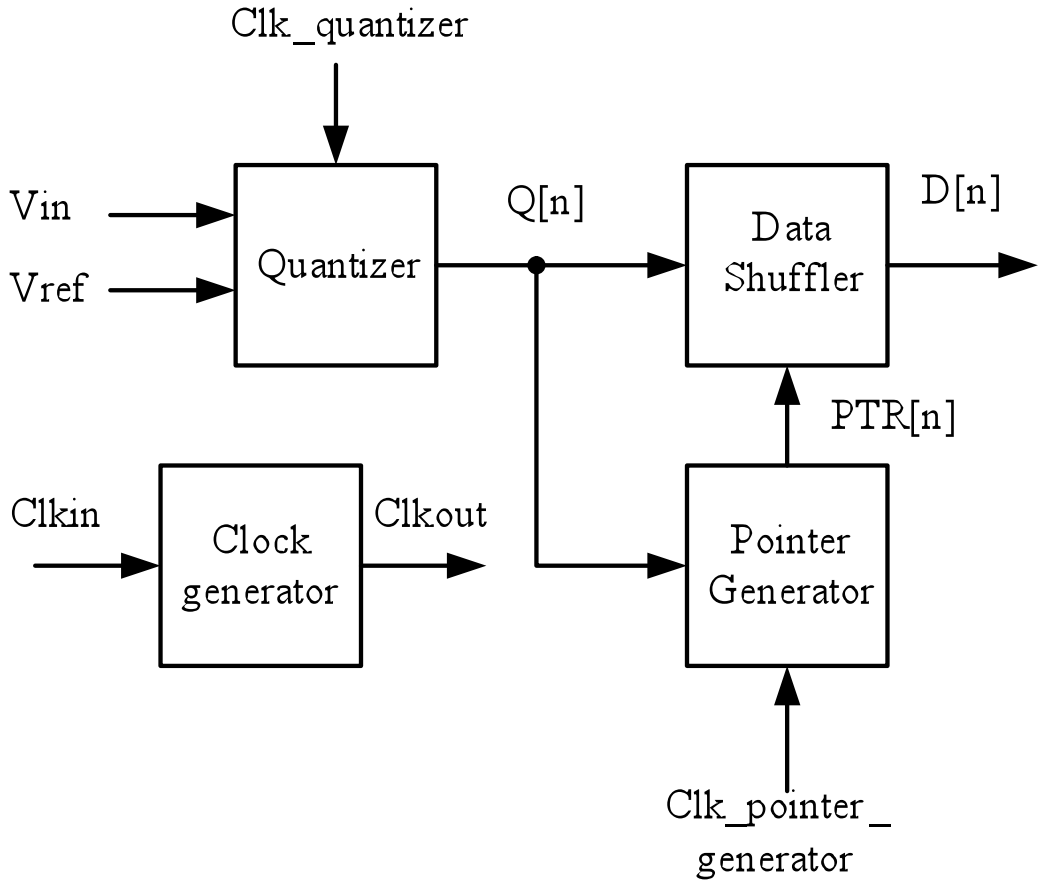


DWA timing conclusion

- In a ΔΣ modulator with distributed feedback:
- Half a clock cycle can be allocated for quantization and data shuffler delay
- A whole clock cycle can be used to calculate new pointer for next shuffler operation
- These features offer both reliability and lower power consumption

DWA simulation

- A typical test bench

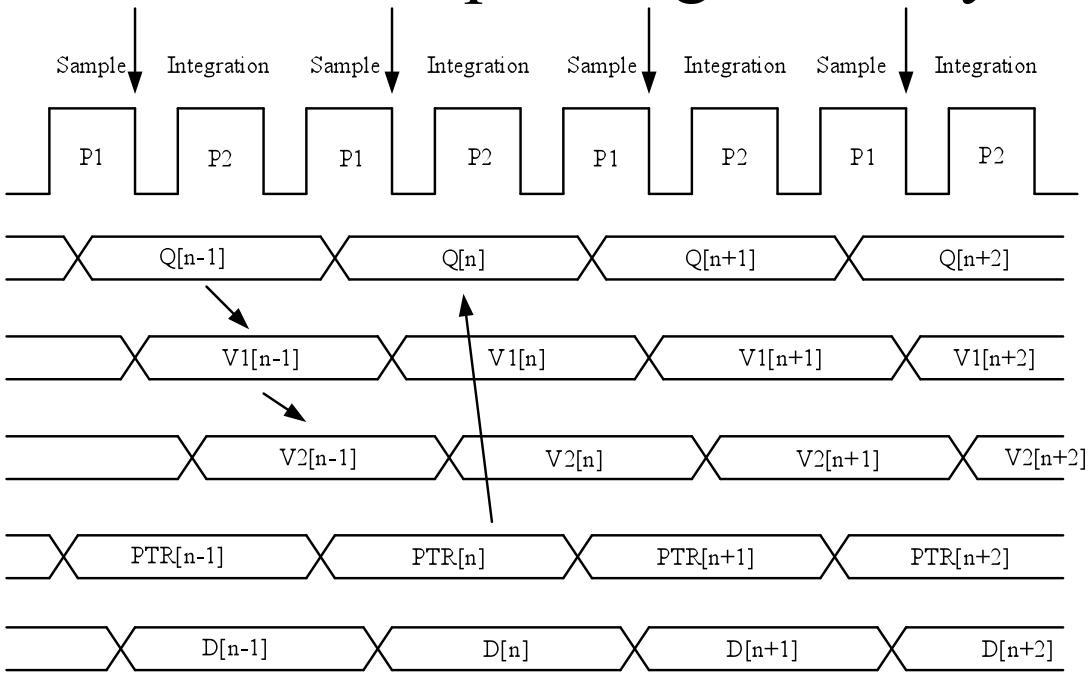


DWA simulation

- Run transient simulation with 1 or 2 cycle of the input signal
- Due to oversampling, this implies a large amount of ADC clock cycles
- Check functionality with PVT variation, usually SS corner with low supply voltage will give a worst case scenario

DWA simulation

- Check eye diagram of important nodes to verify correct operation
- Be sure to run post layout simulation since parasitic capacitance can lower speed significantly



Other DWA implementations

- Bi DWA [2]
- Pseudo DWA [3]
- Partition DWA [4]
- Split set DWA [5]
- Segmented DWA [6]
- DWA can be also performed to shuffler the reference voltages in the quantizer to completely eliminate the delay in the modulator feedback path [7]

References

1. M. Miller et al., “A Multibit Sigma-Delta AD for Multimode Receivers,” *IEEE J. Solid-State Circuits*, vol. 38, pp. 475-482, Mar. 2003.
2. I. Fujimori et al., “A 90dB SNR 2.5-MHz output rate ADC using cascaded multibit delta-sigma modulation at 8× oversampling ratio,” *IEEE J. Solid-State Circuits*, vol. 35, pp. 1820-1827, Dec. 2000.
3. A. A. Hamoui, and K. Martin, “High-order multi-bit modulators and pseudo data-weighted-averaging in low-oversampling ΔΣ ADCs for broad-band applications,” *IEEE T. Circuits and Systems*, vol. 51, pp. 72-85, Jan. 2004.
4. K. Vleugels et al., “A 2.5-V Sigma-Delta Modulator for Broadband Communications Applications,” *IEEE J. Solid-State Circuits*, vol. 36, pp. 1887-1899, Dec. 2001.
5. R Wang and G Temes, “Split Set Data Weighted Averaging,” Electronic Letters Feb. 2006.

References

6. Z. Zhang, and G. Temes, “A Segmented Data Weighted Averaging Technique*,” IEEE ISCAS,* pp. 481-484, May. 2007.
7. L. Dorrer et al., “10-Bit, 3 mW Continuous-Time Delta-Sigma ADC for UMTS in a 0.12µm CMOS Process,” in Proc. Eur. Solid-State Circuits Conf. (ESSCIRC), Lisbon, Portugal, Sept. 2003.