As today’s handsets add features in increasingly complex form factors, serialization techniques are being used to meet handset design goals. Such techniques can allow cost savings with narrower, flexible printed-circuit boards (FPCBs) in addition to reducing PCB space. They also can eliminate unnecessary electromagnetic-interference (EMI) components while improving reliability from the use of smaller connectors. Utilizing serialization also allows designers to meet increasingly smaller and more complex hinge designs by dramatically reducing the number of signals that are sent across the FPCB. Even with all of these benefits, however, some may think that adding devices to implement a serial scheme comes at the cost of increased system power. Yet for handset designers, who are given strict power budgets, power savings can actually be realized through serialization techniques.

A typical parallel implementation is shown in Figure 1. With this architecture, the baseband-processor (BP) drivers see a load that includes the traces of the main PCB, flexible PCB, FPCB connectors, and traces on the flip PCB along with the load of the actual displays. The BP drivers need to be able to drive this load directly using LVCMOS-type signaling. Displays that use an RGB-style interface can require up to 24 bits of data that can run up to 8 MHz for WQVGA or more, depending on the display resolution. As display resolutions increase, the required signal frequency to the display interfaces also rises.

Serial Implementation

In a serial display implementation, a pair of devices is placed in the datapath on the main PCB and flip PCB. The serializer, which is placed near the BP on the main PCB, converts the parallel display data to a serial datastream. That datastream is sent across the FPCB to the deserializer. Depending on the serialization architecture, the number of serial data signals can be reduced to as little as a single differential pair. The deserializer converts the serial datastream back to a parallel datastream, which is driven to the display interface (see Figure 2).

There are key differences between parallel and serial implementations, which allow for link power savings to be achieved with the serial implementation. One key difference is the BP-to-serializer interface. By using a serializer on the main PCB, the BP output drive requirements are significantly reduced. After all, the driven load into the serializer inputs is much lower than that required by the parallel display path. Using a serializer interface also enables the BP to lower the output voltage. In addition, the serialization devices can handle the voltage translation to the display driver. With a display that requires 2.7 V to operate, for example, the BP could drop the voltage to 1.8 V to the serializer. This deserializer would then generate the 2.7-V signaling required for the display.

Most serialization schemes use differential signaling protocols like low-voltage differential signaling (LVDS). This protocol dramatically reduces the swing required to send the data across the FPCB. It also reduces the link’s EMI signature. Power savings can be achieved in the serial implementation because of the reduced signal swing and the elimination of the double-shielded FPCB, which is enabled by the reduction in EMI in the serial stream.
**Power Tipping Point**

The power tipping point is the point at which using a serialization scheme begins to consume less power than the parallel solution in a given application. In this example, the power tipping point is approximated by comparing display link power dissipation in serial versus parallel datapaths in a mobile handset. Practical estimates of system parameters are used. Dynamic power dissipation can be calculated with the following equation:

\[
P_D = \frac{CV^2}{2} \left( F_{CLK} \cdot A_{FACTOR} \cdot N_{BITS} \right)
\]

In this equation,
- \(C\) = the effective load capacitance of the link that’s being driven
- \(V\) = the voltage amplitude of the display signals
- \(F_{CLK}\) = the display path frequency
- \(A_{FACTOR}\) = the effective activity factor of the display signals (average rate of level transitions for the data bits)
- \(N_{BITS}\) = the number of data bits in the display path

This article has discussed most of these parameters except for \(C\) and \(A_{FACTOR}\). \(C\) is system dependent. For a parallel implementation, however, a value of 80 pF is being used to approximate a typical application that includes PCB traces, flex connectors, FPCB, ESD/EMI components, and the display driver’s input load. \(A_{FACTOR}\), on the other hand, is data dependent. It can vary widely depending upon the application. For this calculation, it is assumed to be 50%.

**Figure 3:** This equation can be used to determine dynamic power calculation.

Using these parameters and the equation in Figure 3, the parallel implementation has a power dissipation of 29 mW with the following parameters:

- \(C = 80\) pF
- \(V = 2.7\) V
- \(F_{CLK} = 8\) MHz
- \(A_{FACTOR} = 50\%\)
- \(N_{BITS} = 24\)

For the serial implementation, the power calculation is slightly different. The approach here is to calculate the power dissipation required for both the BP to drive the serializer and the deserializer to drive the display driver. Starting with the BP-to-serializer path, the power dissipation is 0.5 mW based on the following parameters:

- \(C = 3\) pF
- \(V = 1.8\) V
- \(F_{CLK} = 8\) MHz
- \(A_{FACTOR} = 50\%\)
- \(N_{BITS} = 24\)

Significant power savings can obviously be realized through the ability to drop the voltage and drive current of the baseband IO, thanks to the reduced loading of the baseband outputs. Using the same method, the power dissipation in the deserializer-to-display driver path is 14.5 mW based on the following parameters:

- \(C = 40\) pF
- \(V = 2.7\) V
- \(F_{CLK} = 8\) MHz
- \(A_{FACTOR} = 50\%\)
- \(N_{BITS} = 24\)

The power tipping point for this example is thus set at 14 mW, which is the difference between the power in the parallel scheme and the power in the serial scheme’s parallel portions. This power tipping point sets the break-even threshold for serial-link power consumption. Current serial schemes have power-dissipation specifications of 20 mW or less for this example. Adding such schemes therefore adds only 6 mW or less to designs. Serial schemes can actually achieve the power tipping point when further power savings are accounted for, such as the removal of passive components that are often required in parallel solutions.

Based on the exact implementation, further reductions to the display-path power dissipation may be realized by serialization. After all, serialization increases the power tipping point. Such reductions could include removing EMI components like common-mode chokes that may no longer be needed as well as eliminating ESD protection devices on the display path. In the serialization implementation, the serializer and deserializer pair provides ESD isolation for the BP and display driver from transients on the FPCB.

Another way to realize additional power savings with serialization would be to integrate the deserializer into the display driver. This approach, which has been seen in some applications, would significantly reduce the large capacitive load seen by the deserializer while further reducing power. Even in applications that don’t integrate the deserializer, savings can be achieved by allowing closer placement of the deserializer to the display driver. The length and load of the traces in that path will then be reduced.

In conclusion, serialization techniques are becoming increasingly more common in today’s handset designs. Although serialization is primarily used to save space, it is believed to come with a large increase in system power consumption. In reality, however, serialization can actually enable power savings. Current serialization solutions are closing the gap between the serial power dissipation and the power tipping point. In addition to all of the other benefits that serialization adds to the design, serialization solutions are enabling link power savings that can reduce total system power consumption.