A Low-Power Chipset for a Portable Multimedia I/O Terminal

Anantha P. Chandrakasan, Andrew Burstein, and Robert W. Brodersen, *Fellow, IEEE*

***Abstract*— This paper presents the design of a low-power chipset for a portable multimedia terminal that supports pen input, speech I/O, text/graphics output, and one-way full-motion video. Its power consumption was minimized using an approach that involves optimization at all levels of the design, including extended voltage scaling, reduced swing logic, and switched capacitance reduction through operation reduction, choice of number representation, exploitation of signal correlations, self-timing to eliminate glitching, logic design, circuit design, and physical design. The entire chipset, which performs protocol conversion, synchronization, error correction, packetization, buffering, video decompression, and D/A conversion, is implemented in 1.2 μm CMOS and operates from a 1.1 V supply while consuming less than 5 mW.**

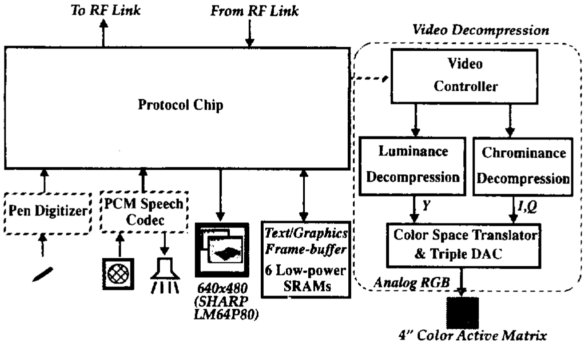


Fig. 1. Overview of the InfoPad Multimedia Terminal.

I. Introduction

THE NEAR future will bring the fusion of three rapidly emerging technologies: personal communications, portable computing, and high bandwidth communications. Over the past several years, the number of personal communications services and technologies has grown explosively. In portable computing, "notebook" computers more powerful than the desktop systems of a few years ago are commonplace. Wide area high bandwidth networks that are in the planning stages will provide the backbone to interconnect multimedia information servers. Wireless communications through radio and IR links are being used as LAN replacement as well as to provide ubiquitous audio communications. However, there has yet to be an integration of these diverse services, in part because of the difficulty in providing a portable terminal that can process the high speed multimedia data provided by the network servers. The circuitry to support the wireless link for such a terminal is being investigated and a low-power solution to this problem is felt to be feasible [1]. The focus of this paper is the processing between this wireless modem and the terminal I/O devices with emphasis on how this processing can be realized with minimal possible power consumption.

II. System Overview of the Portable Multimedia Terminal

The portable multimedia terminal described here provides untethered access to fixed multimedia information servers. The terminal is designed to transmit audio and pen input from the user to the network on a wireless uplink and to receive audio, text/graphics and compressed video from the backbone on the downlink. The portability requirement forces a design focus on power reduction. The availability of communications between the terminal and computational resources on the wired network provides a major degree of freedom for optimizing power; i.e. any computation that does not have to be performed on the terminal, including both general purpose applications and certain I/O tasks such as speech and handwriting recognition, can be moved to the backbone network.

Since the general purpose applications are performed by compute servers on the backbone network, the terminal electronics only has to provide the interface to I/O devices, as shown in Fig. 1. Six chips (a protocol chip, a bank of six 64 kbit SRAM's for text/graphics frame-buffering, and four custom chips for the video decompression) provide the interface between a high speed digital radio modem and a commercial speech codec, pen input circuitry, and LCD panels for text/graphics and full-motion video display. The chips provide protocol conversion, synchronization, error correction, packetization, buffering, video decompression, and digital-to-analog conversion. Through extensive optimization for power reduction, the total power consumption is less than 5 mW.

The protocol chip communicates between the various I/O devices in the system. On the uplink, 4 kbps digitized pen data and 64 kbps, μ-law encoded speech data are buffered using FIFO's, arbitrated and multiplexed, packetized, and transmitted in a serial format to the radio modem. On the down link, serial data from the radio at a 1 Mbs rate is depacketized and demultiplexed, the header information (containing critical information such as data type and length) is

Manuscript received May 16, 1994; revised August 7, 1994. This project was supported by ARPA. The work of A. Chandrakasan was supported by a Fellowship from IBM.

A. P. Chandrakasan is with the Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139 USA.

A. Burstein and R. W. Brodersen are with the EECS Department, Cory Hall, University of California, Berkeley, CA 94720 USA.

IEEE Log Number 9406261.

0018-9200/94$04.00 © 1994 IEEE

error corrected and transferred through FIFO's to one of the three output processing modules: speech, text/graphics, and video decompression.

The speech module reads parallel data from a FIFO and transmits it in a serial format to a codec at a rate of 64 kbs. The text/graphics module handles the protocol used to transmit the bitmaps for text/graphics. The text/graphics module corrects error sensitive information using a Hamming (7, 4, 1) code, and generates the control, timing and buffering for conversion of the 32 bit wide data from the text/graphics frame-buffer (implemented using six low-power 64 kbit SRAM chips) to the 4 bits at 3 MHz required by a 640 x 480, passive LCD display. The final output module is the video decompression, which is realized using four chips. The decompression algorithm is vector quantization, which involves memory lookup operations from a codebook of 256 4 × 4 pixel patterns. Compressed YIQ video is buffered using a ping-pong scheme (one pair of memories for Y and one pair for IQ), providing an asynchronous interface to the radio modem and providing immunity against bursty errors. The amount of RAM required is reduced over conventional frame-buffer schemes by a factor of 24 by storing the video in compressed format. The YIQ decompressed data is sent to another chip that converts this data to digital RGB and then to analog form using a triple DAC that can directly drive a 4 inch active matrix color LCD display. A controller chip performs the video control functions, including the synchronization of the various chips and the color LCD display, control of the ping-pong memories, and loading of the code-books. It uses an addressing scheme that eliminates the need for an output line buffer.

III. Low-Power Implementation of the I/O Processing Modules

The power consumption in CMOS circuits is primarily attributed to charging of load capacitors and is given by:

$$P_{\text{switching}} = \alpha_{0\to 1} C_L V_{dd}^2 f_{\text{clk}} \tag{1}$$

where f_{clk} is the clock frequency, V_{dd} is the operating supply voltage, C_L is output load capacitance, and $\alpha_{0\to 1}$ is the node transition activity factor, the fraction of the time the node makes a power consuming transition (i.e., a $0 \to 1$ transition) inside the clock period. For example, Fig. 2 shows the activity factor, $\alpha_{0\to 1}$, for a 2-input NOR gate as a function of the input signal probabilities, P_a and P_b,

$$\alpha_{0\to 1} = P_0 P_1 = (1-(1-P_a)(1-P_b))(1-P_a)(1-P_b). \tag{2}$$

From this plot, it is clear that understanding the signal statistics and their impact on switching events can be used to significantly reduce the power dissipation. Gate-level power analysis tools have been developed that use probabilistic approaches to estimate the internal node activities of a network given the distribution of the input signals [2],[3]. This chipset exploited signal statistics to reduce the total number of transitions. The physical capacitance is reduced by utilizing a low-power cell library that features minimum sized transistors and optimized layout. In addition to minimizing the switched capacitance, an architecture driven voltage scaling strategy is

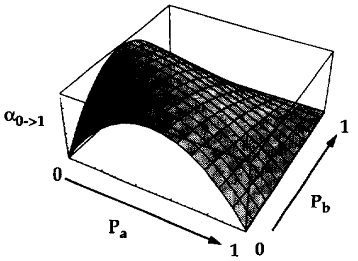


Fig. 2. Influence of signal statistics on switching activity: Transition probability for 2-input NOR gate.

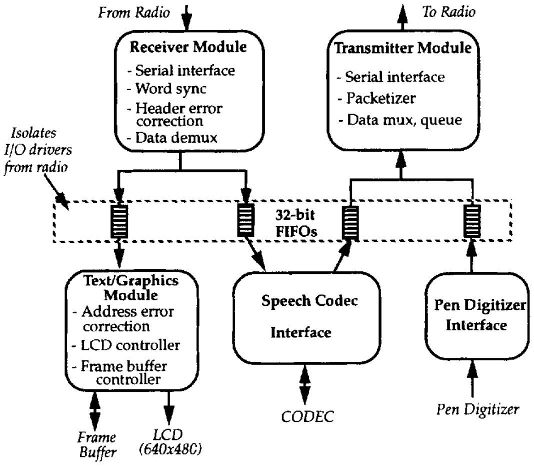


Fig. 3. Block diagram of the protocol chip.

used to scale the supply voltage to the 1–1.5 V range while meeting the computational throughput requirements. Since the circuits operate at a supply voltage that is less than the sum of the NMOS and PMOS threshold voltages ($V_{tn} = 0.7$, $V_{tp} = -0.9$), the devices can never conduct simultaneously for any possible input voltage, *eliminating* short-circuit power. Also, due to the high values of the threshold voltage, the leakage power is negligible.

A. Protocol Module

The protocol chip, shown in Fig. 3, communicates between the various I/O devices in the system. It provides the interface between a commercial radio modem and the pen digitizer, the speech codec, and the 640 x 480 text/graphics LCD display. It also controls the custom frame-buffer of the text/graphics display and provides the necessary interface between the frame-buffer and the LCD display. The wireless modem is isolated from the I/O processing modules using 32-bit asynchronous FIFO's.

1) Parallelism Enables Operation in the 1–1.5 V Range: Since power is proportional to the square of the supply voltage, it is only necessary to reduce the supply voltage for a *quadratic* improvement in the energy consumption. Unfortunately, this simple solution to low power design comes at the cost of increased gate delays and therefore lower

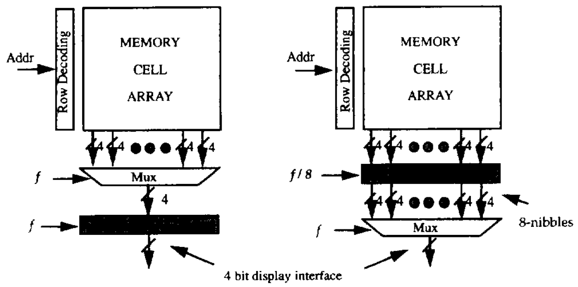


Fig. 4. Parallel memory access enables low-voltage operation.

functional throughput. As presented in [4], an architecture driven voltage scaling strategy using parallelism and pipelining to compensate for the increased gate delays at reduced supply voltages can maintain functional throughput. Parallelism and pipelining were used extensively in this chipset for both arithmetic computation and memory access, enabling supply voltage scaling into the 1–1.5 V range.

To illustrate this strategy, consider the example of reading black and white pixel data from the text/graphics frame-buffer to the 640 × 480 LCD display. The LCD display is a split-panel display in which the top half (640 × 240) and bottom half are addressed in parallel. Each half requires 4 bits (or 4-pixel values) at a rate of 3 MHz. Fig. 4 shows two alternate schemes for reading the required 4 bits of data from memory at the throughput rate, f. On the left side is the serial access scheme in which the 4 bits of data are read out in a serial format and the memory is clocked at the throughput rate f. For this single-ported SRAM frame-buffer implementation (in which the read/write data and address busses are shared) utilizing a serial access scheme requires a supply voltage of 3 V. Another approach (shown on the right side) is to exploit the sequential data access pattern for the frame-buffer and access several 4-bit nibbles in parallel, allowing the memory to be clocked at a lower rate. For example, reading eight nibbles in parallel from the memory allows the memory to be clocked at $f/8$ without loss in throughput. The latched data (8 nibbles) in the protocol chip is then multiplexed out at the throughput rate. This implies that the time available for the memory read operation for the parallel implementation is 8 times longer than the serial scheme and therefore the supply voltage can be dropped for a fixed throughput. The parallel version can run at a supply voltage of 1.1 V (which corresponds to the voltage at which the gate delays increase by a factor of 8 relative to the serial access scheme running at 3 V) while meeting throughput requirements. It is interesting that architectural techniques can be used to drive the supply voltage to such low levels even with a process that has $V_{tn} = 0.7$ V and $V_{tp} = -0.9$ V. If the process could be modified to reduce the threshold voltage, the power supply voltage and therefore the power consumption can be further reduced. An optimum threshold voltage compromises between switching power and leakage power and was found to be around 0.3 V [5].

In general, for signal processing applications that do not have feedback in the computation, arbitrary amount of parallelism can be exploited to reduce the power supply voltage. However, exploiting parallelism often comes at the expense of increased silicon area (for example, there is an increase in the number of I/O pins for the SRAM chip going from a serial memory architecture to a parallel memory access architecture) and capacitance. The increase in capacitance must be traded off against lower voltage to arrive at an optimum level of parallelism and voltage [4]. For DSP applications that have feedback in the computation, algorithmic transformations are required to exploit parallelism and reduce power consumption [6]. Architecture driven voltage scaling can also be applied to general purpose computation (by using superscalar or super-pipelining architectures) but the amount of parallelism in the computation is often limited [7].

2) Reduced Swing Circuitry: Reducing the power supply voltage is clearly a very effective way to reduce the energy per operation since it has a quadratic impact on the power consumption. At a give supply voltage, the output of CMOS logic gates make rail to rail transitions; an approach to reducing the power consumption further is to reduce the voltage swing on large capacitance nodes. For example, using a NMOS device to pull up a node will limit the swing to $V_{dd} - V_t$, rather than rising all the way to the supply voltage. The power consumed for a $0 \rightarrow V_{dd} - V_t$ transition will be $C_L V_{dd}(V_{dd} - V_t)$, and therefore the power consumption reduction over a rail to rail scheme will be $\propto V_{dd}/(V_{dd} - V_t)$. This scheme of using an NMOS device to reduce the swing has two important negative consequences: first, the noise margin for output high (NM_H) is reduced by the amount V_T, which can reduce the margin to 0 V if the supply voltage is set near the sum of the thresholds. Second, since the output does not rise to the upper rail, a static gate connected to the output can consume static power for a high output voltage (since the PMOS of the next stage will be "on"), increasing the effective energy per transition.

Therefore, to utilize the voltage swing reduction, special gates are needed to restore the noise margin to the signal, and eliminate short-circuit currents. These gates require additional devices that will contribute extra parasitic capacitances. Fig. 5 shows a simplifed schematic of such a gate, used in the FIFO memory cells of the low-power cell library used in this chipset [8]. This circuit uses a precharged scheme that clips the voltage of the bit-line (which has several transistors similar to M5 connected to it) to $V_{dd} - V_t$ where $V_t > V_{t0}$ due to the body effect. The devices M1 and M4 precharge the internal node, $\overline{\text{out}}$, to V_{dd}, and the bit-line to $V_{dd} - V_T$. During evaluation (φ ="1"), if V_{in} is high, the bit-line will begin to drop, as shown in the SPICE output next to the schematic. Because the capacitance ratio of the bit-line to the internal node is very large, once the bit-line has dropped roughly 200 mV to sufficiently turn on M3, the internal node quickly drops to the potential of the bit-line, providing signal amplification. Thus, this circuit conserves energy greatly by reducing the voltage swing on the high-capacitance bit-line, and reduces delay by providing signal amplification.

3) Level-Conversion Circuitry to Interface with I/O Devices: Although the protocol chip operates at a supply voltage of 1.1 V, it has to communicate with I/O devices running at higher voltages—for example and text/graphics 640 × 480

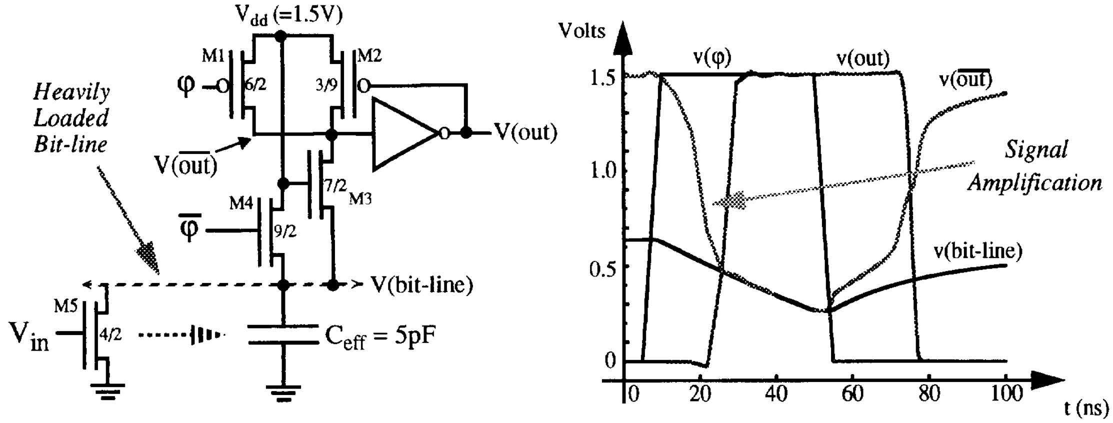


Fig. 5. Signal swing reduction for memory circuits: FIFO example.

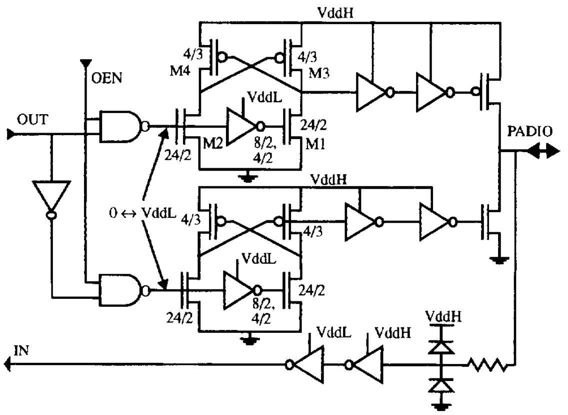


Fig. 6. Level-conversion I/O pad buffer.

LCD display running at 5 V and the speech codec running at 5 V. Level-conversion I/O pads convert the low voltage-swing signals from the core of the low-power chips (e.g., 1.5 V) to the high-voltage signal swings required by I/O devices or vice-versa. Fig. 6 shows the schematic of a level converting I/O pad driver.

This tri-state output buffer uses two supply voltages: the low-voltage supply, V_{ddL}, that is tied to the pad-ring and the high-voltage supply, V_{ddH}, coming the through another unbuffered pad. The low-to-high conversion circuit is a PMOS cross-coupled pair (M3, M4) connected to the high supply voltage, driven differentially (via M1, M2) by the low-voltage signal from the core. The N-device pulldowns, M1 and M2, are DC ratioed against the cross-coupled P-device pullups, M3 and M4, so that a low-swing input ($V_{ddL} = 1$ V) guarantees a correct output transition ($V_{ddH} = 5$ V). That is, the PMOS widths are sized so that the drive capability of the NMOS can overpower the drive of the PMOS, reversing the state of the latch. The ratio is larger than just the ratio of mobilities because the PMOS devices are operating with $V_{GS} = V_{ddH}$, and the NMOS are operating with $V_{GS} = V_{ddL}$. This level-converting pad consumes power only during transitions and consumes no dc power. The remaining buffer stages and output driver and supplied by V_{ddH}. This level-conversion pad will work only with an NWELL process since it requires isolated wells for the PMOS devices that are connected to the high voltage.

4) *Use of Gated Clocks to Reduce the Switching Activity:* At the logic level, gated clocks are used extensively to power down unused modules. For example, consider the error correction in the text/graphics module. The basic protocol for the text/graphics module that is sent over the radio link and through the text/graphics FIFO is a base address for the frame-buffer followed by bit-mapped data—the length information is decoded in the depacketizer module and a control field in the FIFO (a thirty-third bit indicating End of Packet—EOP) is used to delimit the packets. The base address is the starting location in the frame-buffer for the bit-mapped data. While address information is sensitive to channel errors (since errors can cause the data to be written in the wrong area of the screen), bit-mapped data is not very sensitive since errors in data appear as dots on the screen. Therefore, to enable a bandwidth efficient wireless protocol, only the address information is error corrected and the bit-mapped data is left unprotected. Fig. 7 shows a block diagram of a power efficient implementation of this function. A register is introduced at the output of the FIFO and a grated clock enables the error correction module when processing the address information; the ECC is shut down during the rest of the time. The gated clock is generated by a controller which uses the state information of the FIFO (which is based on the $\overline{\text{EMPTY}}$ signal and the "End of Packet" signal from the FIFO). In this manner, the inputs to the ECC are not switching when the data portion of the protocol is accessed from the FIFO. Since typically the address is only a small portion of the text/graphics packet, significant power savings is possible (the amount of power savings is variable depending on packet size). Gated clocks were used in many circuits to tune the frequency and hence clock load for each module.

5) *Low-Power Cell Library:* The entire chipset was designed using a cell library that was optimized for low-power operation. The library contains parameterized datapath cells

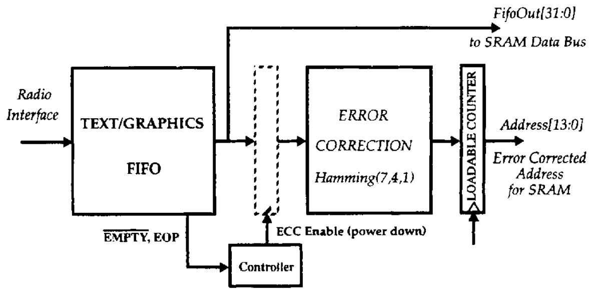


Fig. 7. Gated clocks are used to shut down modules when not used.

(adders, registers, shifters, counters, register files, buffers, etc.), multiple strength and multiple input standard cells for random logic (NAND, NOR, registers, buffers, etc.), low-power memories (FIFO, SRAM), and low-voltage pads (clock pads, low-voltage I/O pads, level-converting pads, etc.). Some of the key design principles used in the cell library are outlined below:

- Device sizing: minimum sized devices were used in most datapath and standard cells. Minimum sized devices should be used to minimize the physical capacitance for circuitry inside the datapath blocks where self-load, rather than interconnect capacitance dominates [4]. When driving large interconnect capacitances, for example when communicating between datapaths, large devices are used.
- Reduced swing circuitry: as described in Section III-A-2, reduced swing circuitry is used limit the voltage swing in memory circuits (FIFO and SRAM) and reduce the power consumption in a linear fashion.
- Self-timed circuits: the memory circuits use self-timing to eliminate spurious transitions on high capacitance data busses. This guarantees the minimum number of transitions per memory access.
- Clocking strategy: the true single phase clock (TSPC) methodology was used for pipeline registers [9]. The schematic along with the device sizes are shown in Fig. 8. The rise-time of the clock signal is critical to proper operation and therefore a control slice attached to each datapath register provides local clock buffering. TSPC was chosen over a two-phase scheme since it reduces the power consumed in the interconnect since only one clock has to be distributed. Fig. 8 also shows a small modification to the TSPC register which can be used by controllers to generate gated clocks. As shown in the timing on the right of Fig. 8, with the extra PMOS device M1, the output (which is the gated clock and is used to clock other registers) can be forced to a ZERO during the low phase of the clock. Without this PMOS device, it will not be possible to generate rising edges for the gated clock on two consecutive rising edges of the system clock.

6) Optimizing Placement and Routing for Low Power: At the layout level, the place and route is optimized such that signals that have high switching activity (such as clocks) are assigned short wires and signals that have low switching activity are allowed to have long wire. Fig. 9 shows an example that involves the routing of large data and control busses from the text/graphics module of the chip core to the pads on the south side of the protocol chip, shown in Fig. 10. The text/graphics module on the protocol chip communicates to both the text/graphics frame-buffer and to the text/graphics 640 × 480 LCD display. The split-panel display requires 8 bits (4 for the top half and 4 for the bottom half) at a rate of 3 MHz, while each SRAM module uses 32 bits clocked at 375 kHz (using the parallel access scheme described in Section III-A-1). An activity factor (for both $0 \rightarrow 1$ and $1 \rightarrow 0$ transitions) of 1/2 is assumed for the data. The address bits are also clocked at 375 kHz but they have a much lower switching activity since the accesses are mostly sequential, coming from the output of a counter. The address bus is time-multiplexed between the read and write addresses for the SRAM, but since the write into the text/graphics frame-buffer is relatively infrequent, the address bus usually carries the read address and is therefore very temporally correlated (the number of transition for a counter output per cycle is $1 + 1/2 + 1/4 + \cdots \approx 2$ since the lsb switches every cycle, the next lsb switches every other cycle, etc.). As seen from the plot of physical capacitance as a function of distance from core to pads, the display data and display clock, which have high switching activity, are assigned the shortest wire lengths, while the SRAM address, which has an activity factor 16 times lower, is allowed to have the longest wires.

B. Text/Graphics Frame-Buffer Module

The frame-buffer for the split panel text/graphics display contains six 64 kbit SRAM chips (Fig. 11), which were synthesized from the low-power cell library's tileable SRAM module. This module meets several constraints to make it a useful component of a wide variety of low power systems.

First, the SRAM module builds memories over a wide variety of sizes. At one extreme, it can create entire memory chips that are only limited by maximum die size, such as the frame-buffer memories used for the text/graphics display. At the other extreme, the same module synthesizes smaller (hundreds or thousands of bit), area and power-efficient memories that are placed on the same chip as datapath and control systems (as in the video decompression chips described in Section III-C). The designer specifies the number of words, which may be any size over two, and the number of bits per word, which may be from two to sixty four. The designer can also control the aspect ratio by specifying the number of blocks at the architecture level, as detailed below.

Next, the SRAM must meet the throughput requirements at reduced supply voltages. The cycle time for an on-chip memory access is 100 ns at 1.5 V in a 1.2 μm process. As described earlier, parallel word access can be used to operate the SRAM at lower voltages without loss in throughput.

The final requirement is to minimize the external timing and control of the SRAM since these signals will appear on the high capacitive external pad and interconnect I/O. The SRAM is synchronized to the rising edge of a single clock; all other timing signals are generated internal to the SRAM, using self-timing circuitry that scales with the size of the SRAM. Also,

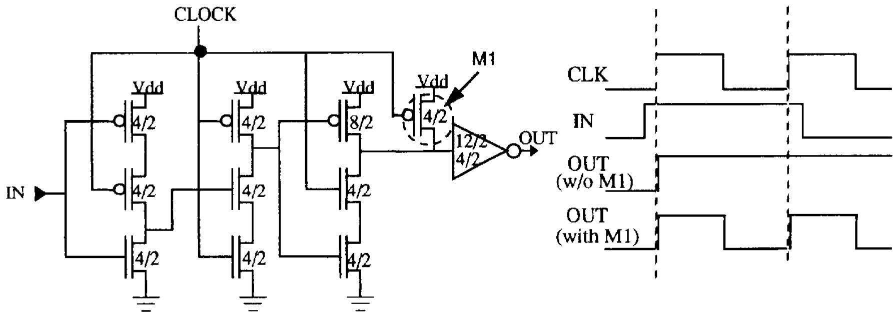


Fig. 8. Schematic of a modified TSPC latch that can be used to generate gated clocks.

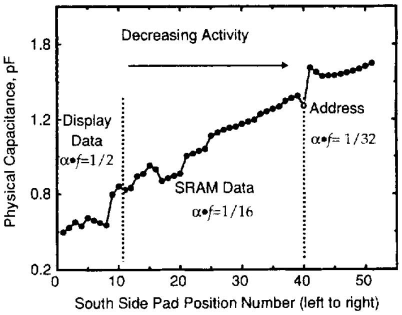


Fig. 9. Optimizing placement for low-power: Example of routing large data/control busses.

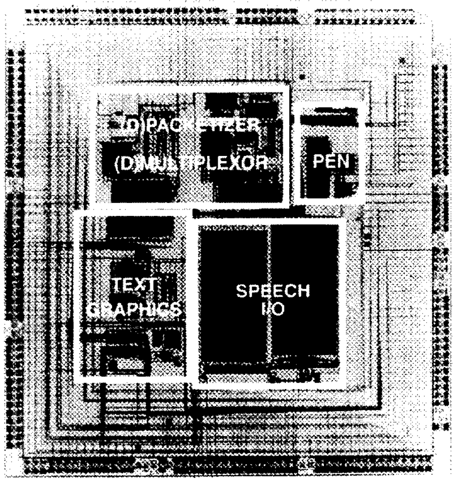


Fig. 10. Die photo of the protocol chip.

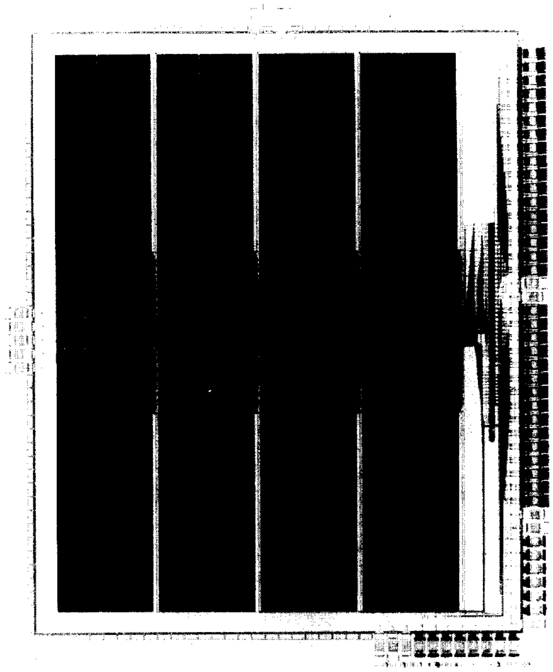


Fig. 11. Die photo of the text/graphics frame-buffer which stores data for the 640 x 480 LCD display.

the SRAM can use either a single, bidirectional bus or can use separate input and output buses.

1) Memory Architecture: The architecture of the SRAM is designed both to minimize power consumption and to enhance scalability. At the highest level of the architecture, the SRAM is organized into a parameterizable number of independent, self-contained blocks, each of which reads and writes the full bit width of the entire memory. For example, the 64 kbit (2 *k* word by 32 bit) frame buffer chips contain eight 8 kbit blocks, each organized as 256 words by 32 bits. Since one of the fundamental power saving techniques is to minimize the effective capacitance switching per clock cycle, the SRAM only activates circuits in one of these blocks per clock cycle (Fig. 12). The power savings groom this architecture are twofold. First, there is less overhead capacitance, since only one set of control signals and decoders switch at one time. Second, and more fundamentally, by having wide data widths into each block, the memory has minimal column decoding, the hence less column capacitance to switch per bit. From an ideal power consumption perspective, it would be best to have no column decoding at all; as a practical concession to pitch matching, this design uses 2-to-1 column decoding.

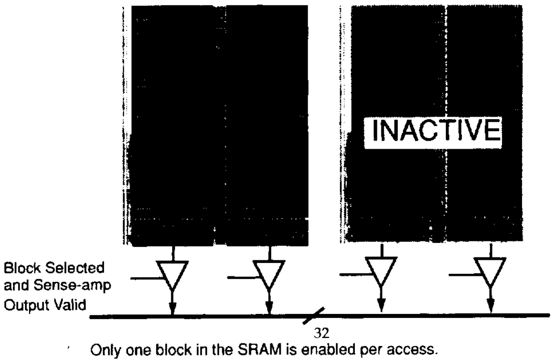

INACTIVE
Block Selected and Sense-amp Output Valid
32
Only one block in the SRAM is enabled per access.

Fig. 12. SRAM block organization.

Power consumption is greatly reduced at the architectural level by beginning each cycle with the block address decoding and then enabling only one block. Memory access would be faster—but power consumption much higher—if all blocks activate at the start of each cycle and the block level decoder selects the output from among all the blocks after the data has been read. In essence, it is not worthwhile to use speculative execution (in this case, speculating on the address) because the gain in speed does not allow enough of a reduction in supply voltage to offset the increase in effective capacitance switched. Controlling the activation of the SRAM at the block level also allows further optimization at the circuit level within the blocks. For example, because no control or clock signals make transitions within a block unless that block is selected, it is often optimal for both speed and power to use fully dynamic circuits; the architecture ensures that dynamic circuits make the same number of transitions as static circuits, so the lower capacitance of the dynamic circuits enables higher speed and lower power. In essence, judicious use of gated clocks (i.e. block level decoding) allowed greater choice of circuit styles.

Using self-contained blocks also makes the architecture flexible and expandable. Since the number of words per block and the total number of blocks are parameterized, the designer can control the aspect ratio by trading off between the number and size of the blocks. Adding blocks to the memory increases its number of words, with minimal effects on power consumption and circuit design. Since circuits in only one block switch at a time (except for the block level decoders, which consume minimal power) the only increase in power consumption is from the increased wiring capacitance between blocks. The only circuits that need to change are the block level decoders. All of the other control, timing, decoding, and sensing circuitry in the blocks are insensitive to the number of blocks in the memory.

2) Circuitry: At the circuit level, the SRAM's most important power saving technique is to reduce the voltage swing on the bitlines. As shown in Fig. 13, the bitlines are precharged through NMOS devices, so the maximum bitline voltage is the apply voltage minus the NMOS threshold voltage (with body effect). Compared to full swing bitlines, this reduces power consumption by only 20% for $V_{dd} = 5$ V, but as much as 50% for $V_{dd} = 1.5$ V and 75% for $V_{dd} = 1.2$ V. (It was not practical to limit the minimum voltage on the bitlines by timing the wordline signals because the timing would have to work for many sized memories, over many voltages, and for different fabrication technologies.) This precharge strategy also allows the column select transistors to double as cascode amplifiers, creating voltage gain between the bitlines and the input to senseamp.

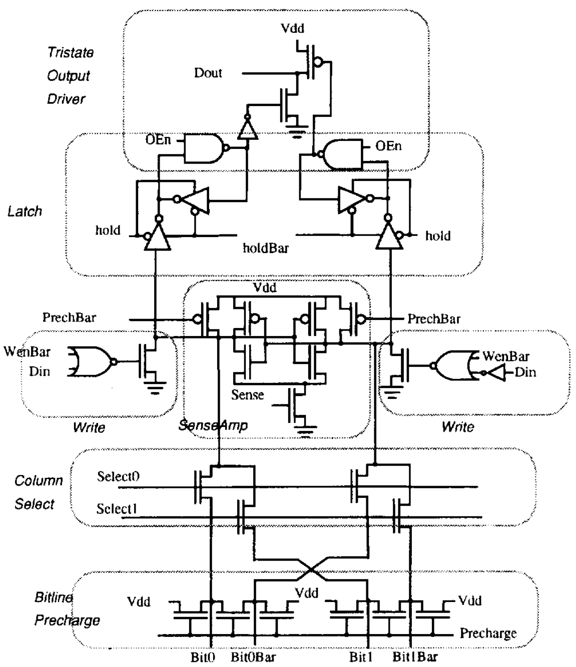


Fig. 13. Precharge, sense-amp, and "glitch"-free tri-state buffer.

Another important power saving technique is to eliminate glitching on the data bus during read operations. The output driver of the senseamp shown in Fig. 13 is tri-stated at the beginning of each clock cycle. Even after the output enable signal is true, the output remains tri-stated until the senseamp's cross-coupled latch has resolved the data value, at which time either the NMOS or the PMOS output device drives the data bus.

One of the major handicaps of using a high threshold process ($V_{tp} = -0.9$ V, $V_{tn} = 0.7$ V) at low supply voltages is that the low gate overdrive voltage for the PMOS ($V_{gs} - V_t$ exacerbates their lower carrier mobility, creating a large imbalance between the current drive of the PMOS compared to the NMOS devices. A partial remedy is to make sure that large capacitances are never driven by more than one series connected PMOS device. The senseamp output driver meets this requirement, in addition to eliminating glitches. Another strategy is to use a non-minimum device ratio ($Wp : Wn$) for circuitry in the critical path.

3) Tiling: To create an SRAM, the designer gives the layout generation tool two parameters—the number of words and the number of bits per word—and may provide and third, optional parameter—the number of blocks. The first step for the layout generator is to tile the individual blocks, each of which has the same layout. Most of the block layout is straightforward: the memory cells are tiled in 2 dimensional arrays (bits, words); the senseamps are tiled in 1 dimensional arrays (words) as are address latches (address lines) and word line buffers (words); a single control slice is added per block. Tiling the address decoding logic is more complex because it employs a tree structure nand decoder. The number of columns in the address decoder is $\log_2$ the number of word lines in the memory cell array. The address decoder contains one row for each word line. However, only the first column contains one transistor for each word line; each subsequent column contains half as many transistors as its predecessor, until the final column has only 2 transistors. Thus, the number of transistors in the decoder tree is approximately twice the number of word lines. The address decoders are placed in the middle of the blocks in order to reduce the RC delay of the poly word lines by a factor of 4.

Between each pair of blocks is an interconnect channel that is also created by tiling. Data, address, control, and power lines are taken from the blocks in first layer metal and connected to second layer metal buses. The interconnect channels are identical except for the block select logic, which decodes the high order address bits to activate a single block. The individual blocks and interconnect channels are tiled together to form the entire SRAM.

C. Video Decompression Module

The final output module is the video decompression module which includes all of the circuitry required to take a compressed video stream from the radio and convert it to the analog data format required by a 4″ (128 × 240 pixels) color active matrix display. This section presents the design of a set of 4 chips (as shown on the block diagram in Fig. 1) to perform video decompression for this low-resolution display; one chip performs all of the control functions for the video decompression and interface to the radio, two chips perform the frame-buffering and the video decompression based on the vector quantization table lookup algorithm, and one chip performs the color space conversion and analog interface to the display.

1) Color (Luminance and Chrominance) Video Decompression: The choice of algorithm is the most highly leveraged decision in meeting the power constraints. The ability for an algorithm to be parallelized is critical and the basic complexity of the computation must be highly optimized. Minimizing the number of operations to perform a given function is critical to minimizing the overall switching activity and therefore the power consumption. The task of selecting the algorithm for the portable terminal depends only on the traditional criteria of achievable compression ratio and the quality of reconstructed images, but also on computational complexity (and hence power), and robustness to high bit error rates.

Most compression standards (for example, JPEG and MPEG) are based upon the Discrete Cosine Transform (DCT). The basic idea in intra-frame schemes such as JPEG is to apply a two-dimensional DCT on a blocked image (typically 8 × 8) followed by quantization to remove correlations within a given frame. In the transform domain, most of the image energy is packed into only a few of the coefficients, and compression is achieved by transmitting only a carefully chosen subset of the coefficients. One main characteristic of the DCT is the symmetric nature of the computation; i.e., the coder and decoder have equal computational complexity. Although the computational complexity of the DCT can be optimized by restructuring algorithms, it still requires several arithmetic and memory operations per pixel [10].

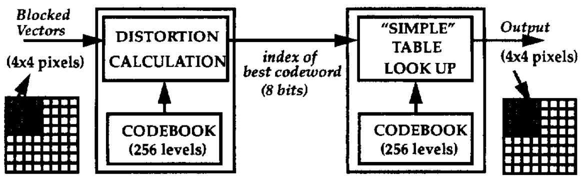


Fig. 14. Video compression/decompression using vector quantization.

An alternative compression scheme is vector quantization (VQ) coding, which is asymmetrical in nature and has been unpopular due to its complex coder requirements. Fig. 14 shows the basic idea behind intraframe vector quantization compression. On the encoder, a group of pixels is blocked into a vector and compared (using some metric such as Mean Square Error) against a set of predetermined reproduction vectors (a set of possible pixel patterns) and the index of the best match is output from the encoder. The decoder has a copy of all possible reproduction vectors (codebook) and the index of the best codeword is used to reconstruct the image using a simple lookup table operation.

For this implementation, the image is segmented into 4 × 4 blocks (i.e., the vector size is 16) and there are 256 entries in the codebook. The original image on the encoder side is represented in the RGB domain using 6 bits for each color plane—using 6 bits to represent video data instead of 8 bits results in very little visual distortion on this low-resolution display. Color information is transformed to the YIQ domain and each plane is individually coded with separate codebooks. The I and Q color components (called the chrominance components) are subsampled in both the horizontal and vertical dimensions. Therefore, the YIQ representation $(Y : 1, I : 1/4, Q : 1/4)$ gives 2 : 1 compression over the RGB representation $(R : 1, G : 1, B : 1)$. On each plane, VQ results in 12 : 1 compression since only 8 bits are transmitted for each 4 × 4 block (choosing 1 out of 256 codes) instead of 16 × 6 bits for the true data. A total of 24 : 1 compression is achieved (32 : 1 if the quantization from 8 bit to 6 bit representation is taken into account) and therefore to support full-motion color video at 30 frames per second, this system consumes a bandwidth of 690 kbits/s on the wireless link.

In addition to the simplicity of the decoder, intraframe VQ has two other advantages. First, VQ localizes errors in space; i.e. errors in the VQ codeword data appear as small corrupted 4 × 4 blocks on the screen and don't corrupt large portions of the screen. Simulations indicate that even with Bit Error Rates as high as 10^{-3}, the image looks reasonable and therefore no error correction was applied on the codebook data, significantly enhancing the bandwidth efficiency of the wireless protocol. Run-length coding, as used in JPEG, is not suitable

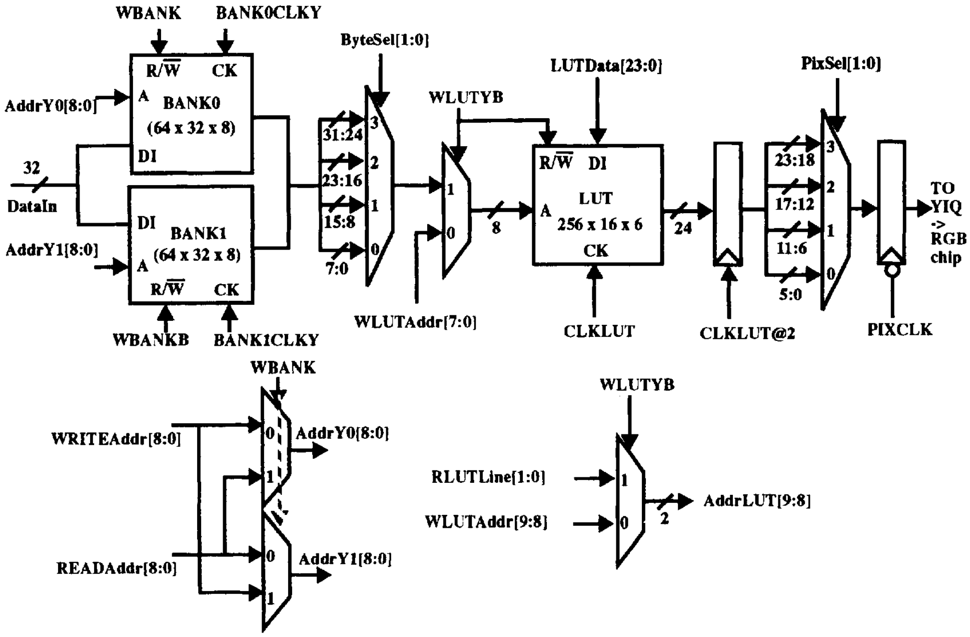


Fig. 15. Block diagram of the luminance decompression chip.

for wireless operation since errors can cause loss of synchronization and can corrupt large portions of the image. Second, intraframe compression localizes errors in time: errors don't accumulate from frame to frame, unlike differential schemes such as MPEG. In differential schemes, errors stay on the screen until a full new image is sent (in an intra frame mode). This chipset does support interframe mode, in which only portions of the screen that are changed are updated. However, this mode can be used only when the bit error rates are low.

Fig. 15 shows a block diagram of the luminance decompression chip. The incoming compressed video data (VQ codewords for the image) is stored using a ping-pong addressing scheme. For example, when the compressed video coming from the RF link is being stored in BANK0, the compressed data stored in BANK1 is read out to be decompressed using the lookup table (LUT). After a full frame of compressed video is assembled in BANK0, the $R/\overline{W}$ signal (signal *WBANK*) of the two frame-buffers toggles, resulting in data being read from BANK0 to the lookup table while new compressed video data is written into BANK1. This ping-pong addressing scheme provides an asynchronous interface between the radio clock and the video system and provides immunity against bursty channel errors; if a frame of compressed video is dropped or if higher priority data is sent over the link (such as text/graphics data or speech which require data to be sent with minimal latency), the complete compressed frame that is already stored in the terminal is decompressed and displayed until a new frame of compressed video is assembled in the other frame-buffer. Also, since the refresh rate of the color display is 60 Hz while the image is updated only at 30 Hz, the ping-pong frame-buffer is actually required unless the bandwidth on the link is doubled.

The video is stored in a compressed rather than uncompressed format to reduce the amount of memory in the system by a factor of 24. Rather than decompressing and storing the data once and displaying the decompressed image twice (since as mentioned above, the refresh rate is 60 Hz while the image is updated at 30 Hz), the image is stored in a compressed format and is read out and decompressed twice; that is, there is no decompressed frame-buffer. The compressed frame-buffer is read out in parallel fashion, where four 8-bit VQ codewords are read in parallel (once again the access pattern of data is know and is exploited) though only one is used at a given time; this once again enabled operation at supply voltage as low as 1.1 V. In this implementation, the video frame-buffer was clocked at 156 kHz while meeting the throughput rate of 2.5 MHz. The output of the frame-buffer is multiplexed at 4 : 1, and is used to index the lookup table which generates the decompressed data.

The chip uses an addressing scheme that eliminates the need for an output line buffer which is typically used to convert a block data format to the raster format required by the display. Fig. 16 shows the numbering of codewords in the frame-buffer and the ordering of pixels inside each block. Each time a codeword is accessed from the frame-buffer, only 4 pixels are read out from the lookup table, creating a raster output which can be sent directly to the display. That is, pixels $P0$–$P3$ corresponding to each codeword between CW0 and CW31 are read from the lookup table and then $P4$–$P7$ are read once again for codewords CW0 through CW31, and then $P8$–$P11$, and finally $P12$–$P15$. Thus, each codeword is accessed four times per image. This approach increases the frequency of codeword access relative to a scheme which reads each codeword only once (and stores $P0$–$P15$ in a line-

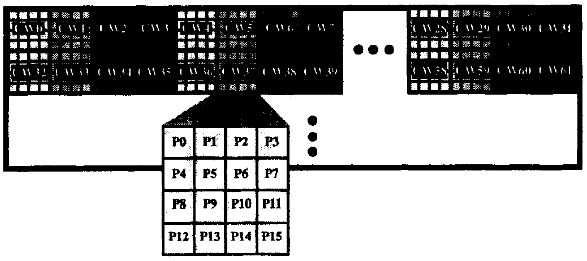


Fig. 16. Addressing of the frame-buffers and lookup table.

buffer), but since the codewords are accessed at a much lower frequently relative to the lookup table data, and since the line-buffer access has been eliminated, the overall power due to memory access is reduced by approximately a factor of 1.5. This is an example of architectural restructuring to reduce the number of operations.

The control of the luminance decompression chip (address generation, clock generation, multiplexor selection, etc.) is performed by the video controller chip. The multiplexors for read/write address lines are integrated in the decompression chip. RLUT Line [1 : 0] controls the row of pixels that are read for a given codeword. The lookup table is programmable over the radio link through the video controller chip. The color video decompression chip is implemented using a very similar architecture to the luminance architecture shown in Fig. 15. Since I and Q data are each sub-sampled by a factor 4, the amount of frame-buffer memory required is reduced by a factor of 4 for each component. Also, since they use the same addressing scheme, the I and Q data are stored in the same physical frame-buffer and the data is interleaved; i.e. the 32-bit frame-buffer word is organized as: CWI0, CWI1 CWQ0 CWQ1, and the 4 : 1 mux to select the codeword in the Y decompression chip is replaced by two 2 : 1 muxes, one for I (to select between CWI0 and CWI1) and one for Q (to select between CWQ0 and CWQ1). The chrominance (I) decompression chip also has two separate ($256 \times 16 \times 6$) lookup table memories. Fig. 17 shows the die photograph of the luminance decompression chip. The chip consumes only 115 μW running at 1.5 V to support a video throughput rate of 2.5 MHz for the 4″ color display.

2) Video Controller: The video controller performs all the control functions for the video decompression module. It generates all of the timing for the NTSC display, interfaces to the radio, controls the frame-buffers and lookup tables and performs synchronization for the system. A summary of the functions performed is outlined below:

- NTSC sync generation for the 4″ display: The LQ4RA01 4″ color display is responsive to a standard composite sync signal with negative polarity of the same amplitude level as that of the video composite signal. The standard sync found in NTSC format has extra timing information such as an extra half line in one field (to distinguish between even and odd fields), and pre and post equalization half-line pulses during vertical sync that are not a necessity for proper operation of the LCD. A significant simplification of this protocol which involves block sync (with no equalization pulses) and the elimination of the half-line can be used to obtain a sync that still provides adequate synchronization information. The implemented sync signal is simply a scaled digital combination of VSYNC (vertical sync) and HSYNC (horizontal sync). The original and modified sync signals are shown in Fig. 18.
- Decodes and demultiplexes the radio data: similar to the function performed in the protocol chip (Section III-A), the video controller chip decodes packets from the radio and demultiplexes between a frame-buffer FIFO and LUT FIFO. Contained inside each frame-buffer packet (that is sent over the frame-buffer FIFO) is the encoding of type information (TYPE = 0 ⟹ data is for the Y frame-buffer, and TYPE = 1 ⟹ data is for the IQ frame-buffer). Similarly, inside the LUT packet is encoding information about the LUT data type (Y, I or Q).
- Controls reading and wiring of the frame-buffer memories for both Y and I decompression chips: it generates the read and write addressed for the ping-pong frame-buffers. It also generates the multiplexor control signals that selects the output of the frame-buffers (ByteSel in Fig. 15). It also performs the R/$\overline{\text{W}}$ control (i.e., controls when the frame-buffers switch between reading to writing) and generates the clocks for the SRAMs.
- Controls loading and reading of the lookup table memories for Y, I and Q data: it generates the write address (WLUTAddr in Fig. 15) and part of the read address (RLUTLine in Fig. 15) for the lookup table memories. It also generates the multiplexor control signals that selects the output of the LUT (PixSel in Fig. 15).

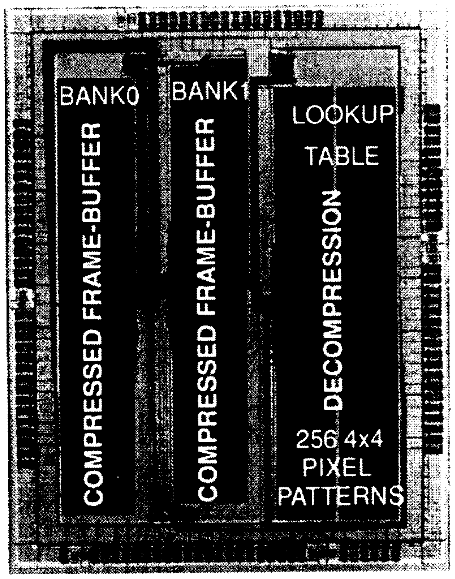
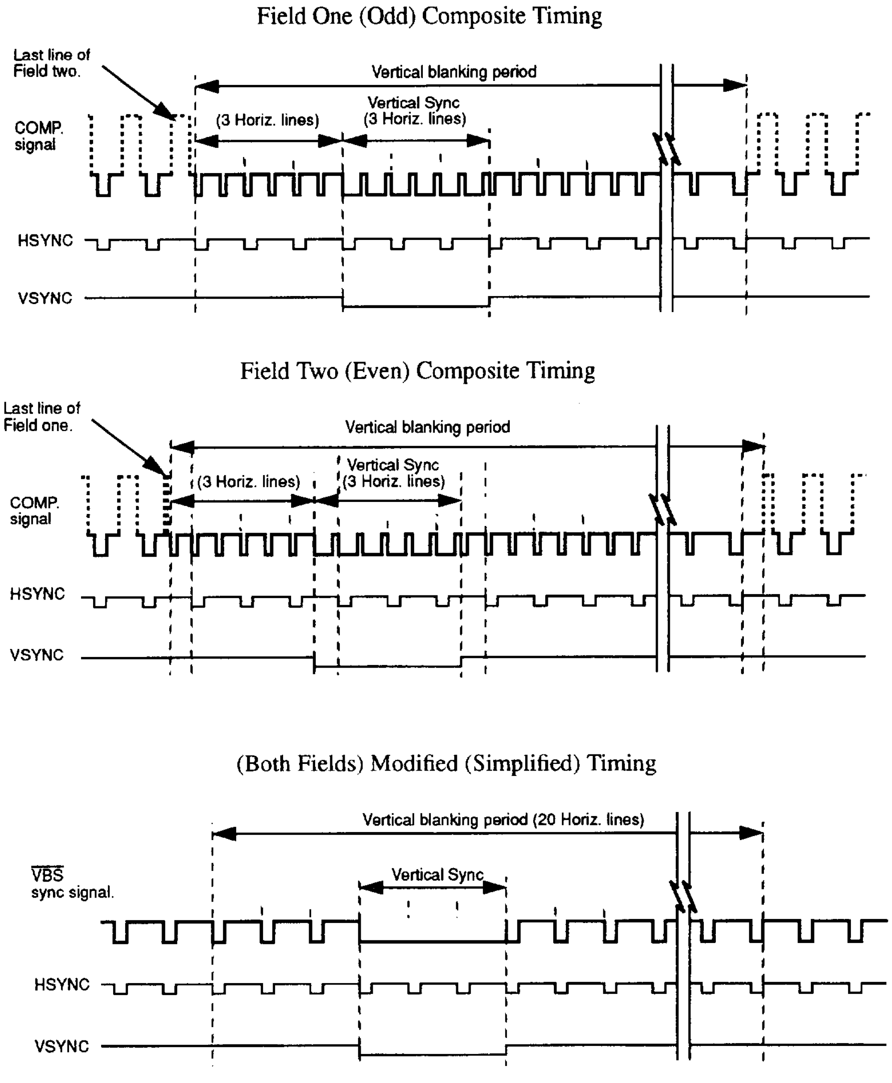


Fig. 17. Die photo of the luminance decompression chip.


Field One (Odd) Composite Timing

Field Two (Even) Composite Timing

(Both Fields) Modified (Simplified) Timing

Fig. 18. Simplification of the NTSC system timing.

- Support for variable sized packets: a decompressed frame of video can be broken into multiple packets and the size of the packets is variable, providing a flexible platform to test the effects of packet sizes. Also, this allows a form of interframe coding in which only the differences between the current frame and the frame corresponding to two image ago is sent. This is effective only if the BER is fairly low for the reasons explained earlier.
- Matches pipelined delays in the system: since the system is pipeline, the output sync signals for the display must be delayed to avoid offsets of pixel data on the display. The timing signals needed by the color space converter are also generated in this chip.

3) Color-Space Converter and Digital-to-Analog Converter: The digital YIQ information from the video decompression chips are sent to a color space converter which converts it to analog RGB to drive the 4″ color LCD display from SHARP. The digital YIQ is first converted to digital RGB and then a triple digital-to-analog converter directly drives the display.

a) Digital YIQ-to-digital RGB conversion: In the YIQ to RGB translation, which involves multiplication with constant coefficients, the switching events are minimized at the algorithmic level by substituting multiplications with hardwired shift–add operations (in which the shift operations degenerated to wiring) and by optimal scaling coefficients. In this way, the 3 x 3 matrix conversion operation degenerated to 8 addition. The implementation was fully parallel and therefore there was no controller. For I/O communication (between the decompression chips and the color space chip) and in the matrix computation, sign-magnitude representation is chosen over two's complement to reduce the toggle activity in the sign bits [11]. For example, going from -1 to 0 with result in all the bits toggling in two's complement representation.

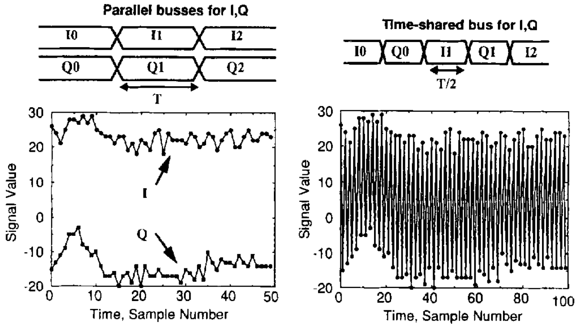


Fig. 19. Time-multiplexing can increase the switching activity.

At the architecture level, time-multiplexing was avoided as it can destroy signal correlations, increasing the activity. Fig. 19 shows two alternate schemes for transmitting the I and Q data from the decompression chips to the color space converter chip. On the left is a fully parallel version in which I and Q have separate data busses. Also shown is the data for I and Q for a short segment in time. As shown, the data is slowly varying and therefore has low switching activity in the higher order bits. On the right is a time-multiplexed version in which there is a single time-shared bus in which the I and Q samples are interleaved. As seen from the signal value on the data bus, there is high switching activity resulting in higher power.

This digital YIQ to digital RGB implementation consumes 100 μW, which is two to three orders of magnitude lower than a design which is not optimized for power. This difference in power is attributed to 1.5 V operation instead of 5 V operation (which provides a ×11 reduction in power), using hardwired optimized shift–add multipliers instead of true-multipliers (which reduces switched capacitance by ×5), using an optimized cell library that uses minimum sized devices and single phase clock methodology (which reduces switched capacitance by ×2–3), using an optimized number representation in I/O and matrix multiplication (×1.2), integrating the DAC with the signal processing (so that the output interchip power is eliminated and reduces power by ×1.3), using an hardware assignment which keeps uncorrelated data on different units (×1.5), and using 6 bits/plane instead of 8 bits/plane (×1.3).

b) Low-voltage digital-to-analog converter: A low-voltage, low-throughput 6-bit DAC has been developed to drive the 4" SHARP LCD. The LCD display takes pixel data in the analog R, G, B format which has voltage levels compatible with the NTSC format (i.e., V_{pp} = 0 to 0.7 V). The DAC, shown in Fig. 20, has an architecture based on a conventional nonweighted current switched array [12]. Based on a decoded 6-bit digital word, an appropriate number of current sources are turned ON and summed. The output voltage is obtained by passing current through an external resistor. Since the settling time requirement is quite low and since the LCD display has a high impedance capacitive load, the external resistor was chosen to be approximately 1 kΩ rather than 75 Ω. This reduces the power consumption by more than an order of magnitude since the average current drawn from the supply is reduced by more than order to magnitude. The row decoding and column decoding logic is identical to the one presented in [12]. The decoding logic was implemented using minimum sized low-power standard cells.

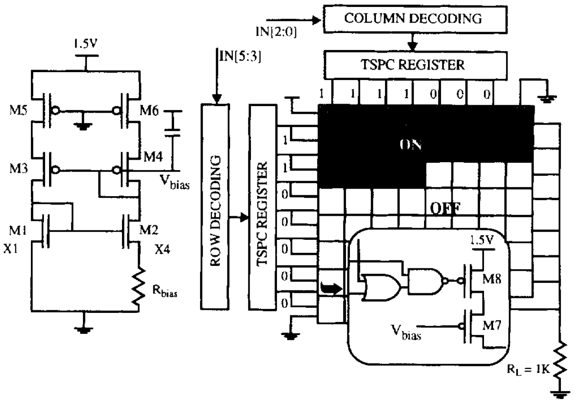


Fig. 20. Low-voltage 6–bit digital-to-analog converter.

TABLE I
SUMMARY OF CHIPSET FOR THE INFOPAD TERMINAL

Chip Description	Area (mmxmm)	Minimum Supply Voltage	Power at 1.5V
Protocol	9.4 x 9.1	1.1V	1.9mW
Frame-buffer SRAM (with loading)	7.8 x 6.5	1.1V	500µW
Video Controller	6.7 x 6.4	1.1V	150µW
Luminance Decompression	8.5 x 6.7	1.1V	115µW
Chrominance Decompression	8.5 x 9.0	1.1V	100µW
Color Space Conversion and Triple DAC	4.1 x 4.7	1.3V	1.1mW

A current source that operates at a reduced supply voltage of 2.7 V has been developed [13]. The current source used in this DAC is similar. However, due to the modest DAC throughput requirements for this 4" display, and to statistically reduce the power consumption by a factor of 2 (on average only half of the current sources will be ON) a single ended architecture is used instead of the differential scheme. The current cell consists of stacked PMOS devices (M7 and M8) with the top transistor M8 being digitally switched, thus operating in the linear region. Therefore, effectively, the output resistance of the current source is the output resistance of a single transistor (M7) degenerated with a source resistor. To increase the output resistance of M7, the length of the device was made nonminimum. In order to operate the DAC down to a supply voltage of 1.5 V and to meet the 0.7 V_{pp} requirement for the output, the W/L of M8 was made large to keep the voltage dropped across M8 to less than 100 mV. Also, the bias voltage was set up so that the $V_{gs} - V_t$ for M7 was approximately 200 mV, allowing supply voltages to be as low as 1.2–1.3 V. The power consumption of the DAC

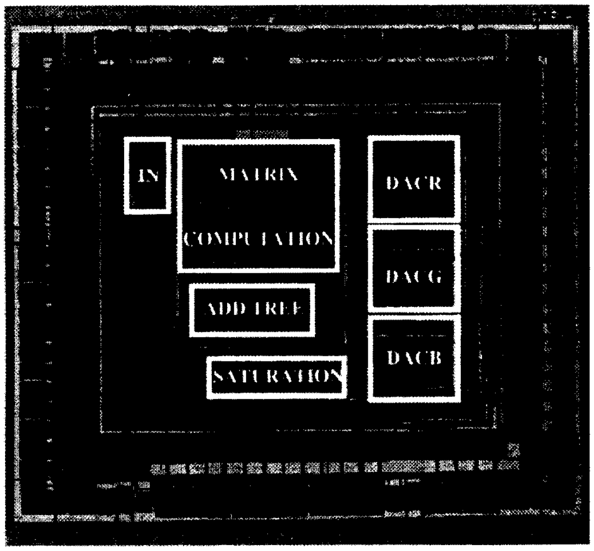
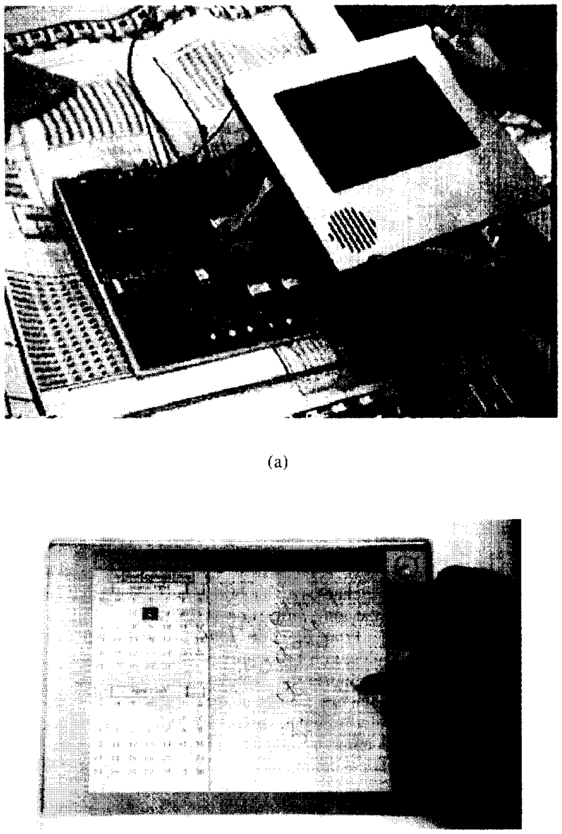


Fig. 21. Die photo of the color-space converter and digital-to-analog converter.



(a)

(b)

Fig. 22. (a) The IPGraphics terminal (pen, speech I/O, and text/graphics). (b) A notebook application running on the IPGraphics terminal.

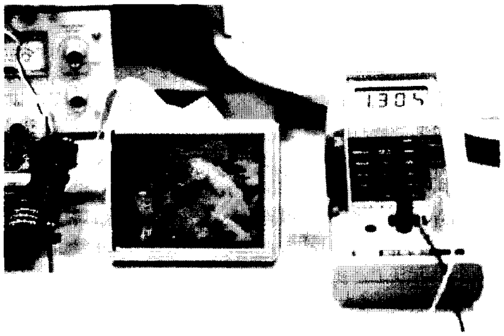


Fig. 23. Output of video decompression chips running at a supply voltage of 1.3 V.

is dominated by the analog power and for a 1.5 V supply: $P_{\text{avg}} = V_{dd} \bullet I_{\text{avg}} = 1.5 \bullet 1/2 \bullet 0.7/1.2\ \text{K} = 440\ \mu\text{W}$. The power measured in the actual system was lower since the DAC was shut down (the digital input was forced to 0) during the horizontal and vertical blanking periods.

The digital decoding inside each current cell is implemented using a single complex CMOS gate $\overline{(\text{A} + \text{B}) \bullet \text{C}}$ to eliminate glitching on the gate of M8 as compared to path balancing approaches [14]. The device sizes on the complex gate are minimum to minimize the dynamic power consumption.

Fig. 20 also shows the schematic of the bias circuitry which is a bootstrapped current reference [15]. The top current mirror, M3 through M6, forces the bias currents in M1 and M2 to be equal. Transistors M5 and M6 operate in the linear region and emulate the voltage drop across the switch transistor in the current cell array (M8). Simple first-order analysis shows that the reference current is given by:

$$I_{\text{ref}} = \frac{2}{K_n R_{\text{bias}}^2} \left(\frac{1}{\sqrt{\frac{W}{L1}}} - \frac{1}{\sqrt{\frac{W}{L2}}} \right)^2 . \tag{3}$$

Note that the reference current ($I_{\text{ref}} = I_1 = I_2$) is to first order independent of supply variations (this is due to the bootstapped techniques used). The bias current is set by sizing M1 and M2 and by choosing R_{bias}. The current source can operate at low supply voltages even with a process that has a standard threshold voltage. The minimum operating voltage for this current reference is given by:

$$V_{dd\min} = V_{ds6} + |V_{tp4}| + V_{d\text{sat}4} + V_{d\text{sat}2} + I_{\text{ref}} R_{\text{bias}}. \tag{4}$$

In the above equation, V_{ds6} is very small through device sizing (<100 mV since M6 is in the linear region), V_{tp} for the MOSIS 1.2 μm CMOS process is typically 0.9 V, and the last three terms can be made small (100–200 mV) by device sizing and choice of bias current. Therefore, the current source can operate down to the 1.2–1.5 V range.

Fig. 21 shows the die photo of the color space translator and triple DAC. Table I shows the summary of the chipset implemented in 1.2 μm CMOS technology.

IV. System Implementation

A PCB containing the protocol chip, 6 SRAM chips, a speech codec, and pen interface logic has been fabricated and tested. Various power supply voltages needed for the design including −17 V (adjustable using a trimpot) for display drive, 12 V for dc-to-ac inverter for the backlight, 1.5 V for the custom chips, and −5 V for the speech codec have been realized using commercial chips. Custom low-voltage high efficiency switching regulators have been fabricated and will be integrated into the next generation terminal [16]. This board is integrated with a commercial radio modem to realize a complete I/O terminal with a 1 Mbit/s wireless channel. Fig. 22 shows the photograph of this terminal, IPGraphics, which supports pen input, speech I/O and text/graphics output. The next version will also provide support for one-way full motion video and Fig. 23 shows the output of the decompression chips (running at 1.3 V) on a 4" SHARP active matrix display.

V. Conclusion

A low-power chipset for a portable multimedia terminal which supports pen input, speech I/O and one-way full-motion video has been presented. A system level approach which involves optimizing the circuitry, the architectures and the algorithms was used to minimize power consumption. The biggest power savings came from an architecture driven voltage scaling strategy that allowed supply voltages as low as lV. The memory circuits used reduced swing circuits that clamped the voltage swings further, resulting in further power reduction. The capacitance switched was minimized through operation reduction, choice of number representation to statistically reduce the number of transitions, exploitation of signal correlations, self-timing to eliminate glitching, use of gated clocks, optimized transistor sizing, and optimizing place and route. The entire chipset that performs protocol conversion, synchronization, error correction, packetization, buffering, video decompression and D/A conversion operates from a 1.1 V supply and consumes less than 5 mW.

Acknowledgment

The authors thank R. Allmon, T. Burd, S. Narayanaswamy, R. Neff, I. O'Donnell, and B. Richards for their invaluable contributions. The mechanical packing for the terminal was designed by R. Doering, C. Smith, and Prof. P. Wright.

References

[1] S. Sheng, A. P. Chandrakasan, and R. W. Brodersen, "A portable multimedia terminal," *IEEE Commun. Mag.*, vol. 30, pp. 64–75, Dec. 1992.

[2] F. Najm, "Transition density, a stochastic measure of activities in digital circuits," *Proc. of DAC*, pp. 644–649, 1991.

[3] A. Ghosh, S. Devedas, K. Keutzer, and J. White, "Estimation of average switching activity," *Proc. DAC*, pp. 253–259, 1992.

[4] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power digital CMOS design," *IEEE J. Solid-State Circ.*, vol. 27, pp. 473–484, Apr. 1992.

[5] A. P. Chandrakasan and R. W. Brodersen, "Minimizing power consumption in digital CMOS circuits," *Proc. IEEE*, Dec. 1994, to be published.

[6] A. P. Chandrakasan, M. Potkonjak, J. Rabaey, and R. Brodersen, "HYPER-LP: A design system for optimizing power using architectural transformations," in *IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1992, pp. 300–303.

[7] D. A. Patterson and J. L. Hennessy, *Computer Architecture: A Quantitative Approach.* San Mateo, CA: Morgan Kaufman, 1989.

[8] T. Burd, "Low-power CMOS library design methodology," Masters thesis, ERL Univ. Calif., Berkeley, Aug. 1994.

[9] J. Yuan and C. Svensson, "High-speed CMOS circuit technique," *IEEE Solid-State Circ.*, vol. 24, pp. 62–70, Feb. 1989.

[10] K. R. Rao and P. Yip, *Discrete Cosine Transform.* New York: Academic, 1990.

[11] A. P. Chandrakasan, R. Allmon, A. Stratakos, and R. W. Brodersen, "Design of portable systems," in *Custom Integrat. Circuits Conf.*, May 1994, pp. 259–266.

[12] T. Miki, Y. Nakamura, M. Nakaya, S. Asai, Y. Akasaka, and Y. Horiba, "An 80-MHz 8-bit CMOS D/A converter," *IEEE J. Solid-State Circ.*, vol. SC-21, pp. 983–988, Dec. 1986.

[13] T. Miki, Y. Nakamura, Y. Nishikawa, K. Okada, and Y. Horiba, "A 10 bit 50 MS/s CMOS D/A converter with 2.7 V power supply," in *1992 Symp. VLSI Circuits*, 1992, pp. 92–93.

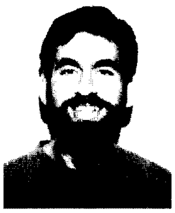
[14] J. M. Fournier and P. Senn, "A 130 MHz 8-b CMOS video DAC for HDTV applications," *IEEE J. Solid-State Circ.*, vol. 26, pp. 1073–1077, July 1991.

[15] P. Gray and R. Meyer, *Analysis and Design of Analog Integrated Circuits*, second ed. New York: Wiley, 1984.

[16] A. Stratakos, S. Sanders, and R. Brodersen, "A low-voltage CMOS DC-DC converter for a portable battery-operated system," in *Proc. IEEE Power Electron. Specialists Conf.*, 1994, pp. 619–626.

Anantha P. Chandrakasan received the B.S. (with highest honors), M.S., and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1989, 1990, and 1994, respectively. He received a Fellowship in Engineering from IBM in 1993.

Since September 1994, he has been an Assistant Professor of Electrical Engineering at the Massachusetts Institute to Technology, Cambridge. He has been awarded the Analog Devices Career Development Chair. His research interests include design methodologies for low-power portable systems, computer-aided design tools for VLSI, and system-level integration.



Andrew Burstein received the B.S. degree from Cornell University, Ithaca, NY, in 1986, and the M.S. degree from the University of California, Berkeley, in 1987, both in electrical engineering.

He has previously worked at Bellcore, and is currently doing his Doctoral dissertation in low-power hardware for automatic speech recognition.

Robert W. Brodersen (F'92) received the B.S. degree in electrical engineering and the B.S. degree in mathematics from California State Polytechnic University. He later received the Engineer's and M.S. degrees from the Massachusetts Institute of Technology, Cambridge, and the Ph.D degree from the same institution in 1972.

From 1972 to 1976, he was with the Central Research Laboratory at Texas Instruments. In 1976, he joined the Electrical Engineering and Computer Science faculty of the University of California at Berkeley, where he is now a Professor. In addition to teaching, he is involved in research into new applications of integrated circuits, focused in the areas of low power design and wireless communications, and the CAD tools necessary to support this activity.

Dr. Brodersen has won best paper awards for a number of journal and conference papers in the areas of integrated circuit design, CAD, and communications, including the W. G. Baker award for the best IEEE publication in all areas, in 1979. In 1983 he was co-recipient of the IEEE Morris Liebmann award. In 1986 he received the Technical Achievement award in the IEEE Circuits and Systems Society, and in 1991 from the Signal Processing Society. In 1988 he was elected a member of the National Academy of Engineering.