

Data Logging Solution for Digital Signal Processors

Brian Newberry

Nekton Research, Inc.

BNewberry@NektonResearch.com

James M. Conrad

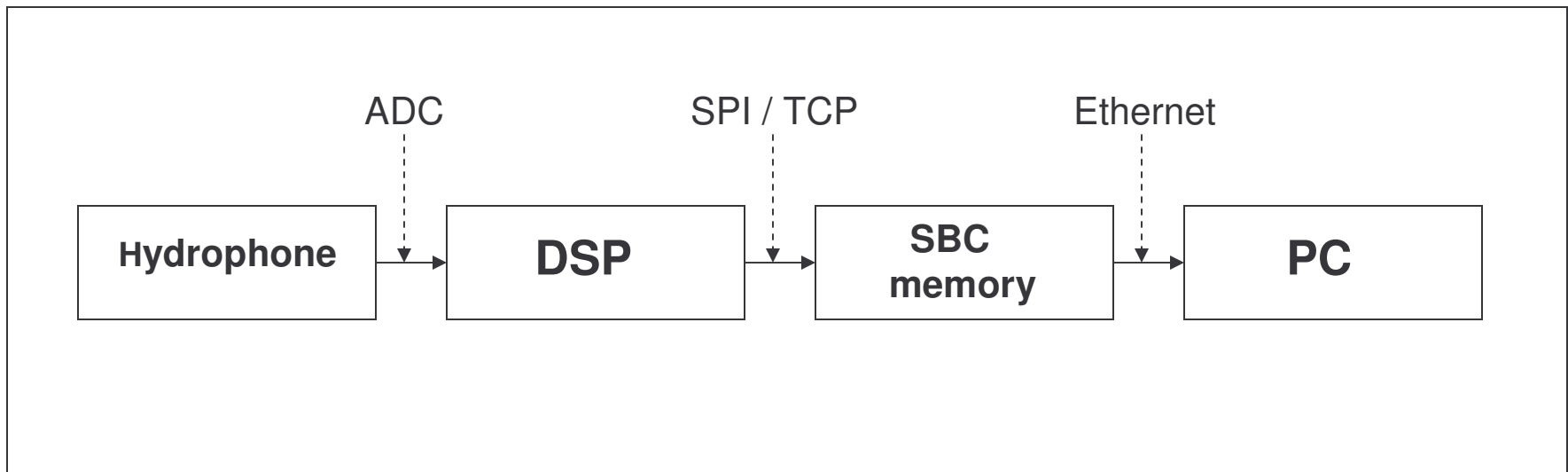
University of North Carolina at Charlotte

jmconrad@uncc.edu

Introduction

- q Analyzing analog data collected by AUV (Autonomous Underwater Vehicle)
 - q Embedded DSP Data Logger Design for audio signal
 - q Digital Signal Processor (DSP) – multiple instructions per cycle
 - q DSP can process audio and video signals with high data rates
 - q DSPs consume low power
-

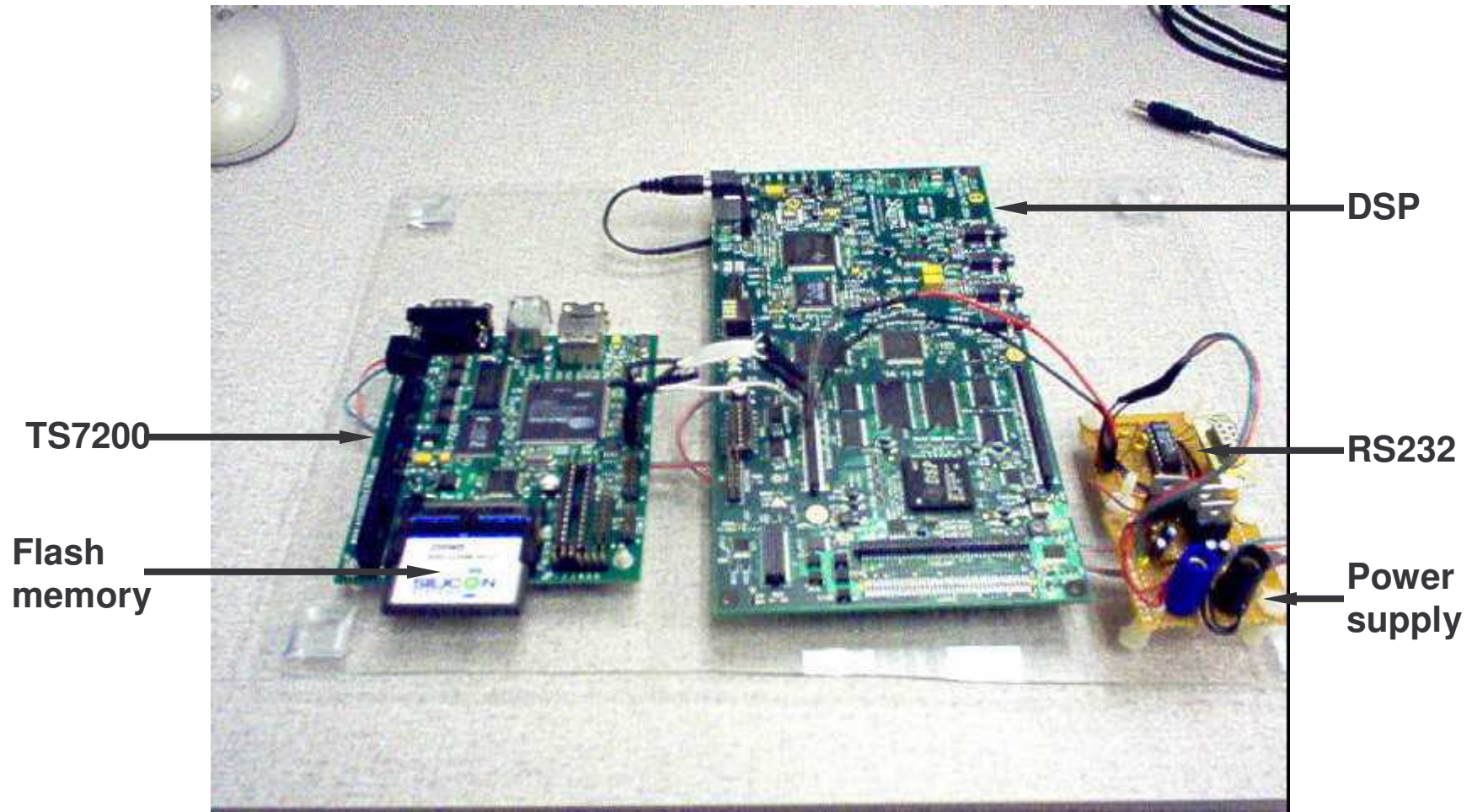
Block Diagram



System Design

Component	Specification
DSP – DSK6416T	1 MB RAM (onboard), 16 MB RAM (offboard), 512 KB Flash
ADC (onboard)	96 kHz/s – sample rate (100 kHz required)
Compact Flash Memory Card	256 MB, low power, 22 minutes data can be stored one time
TS-7200 Compact Flash Controller	Single board ARM based Linux machine, Data rate- 1 Mbit/s, 1 mW power consumption, OS- 5 MB RAM, data rate 0.85 MB/s, FTP (File Transfer Protocol) server
RS232 (Serial Communication)	DSP communicates with hyperterminal

Final System Set-up



Pseudo code (DSP operation)

Initialize system

Wait till told to start

While not told to quit

{

Gather audio data from codec

Process audio data

Report results

Send audio data to TS-7200 over SPI

Wait till reaction to reported results is complete

}

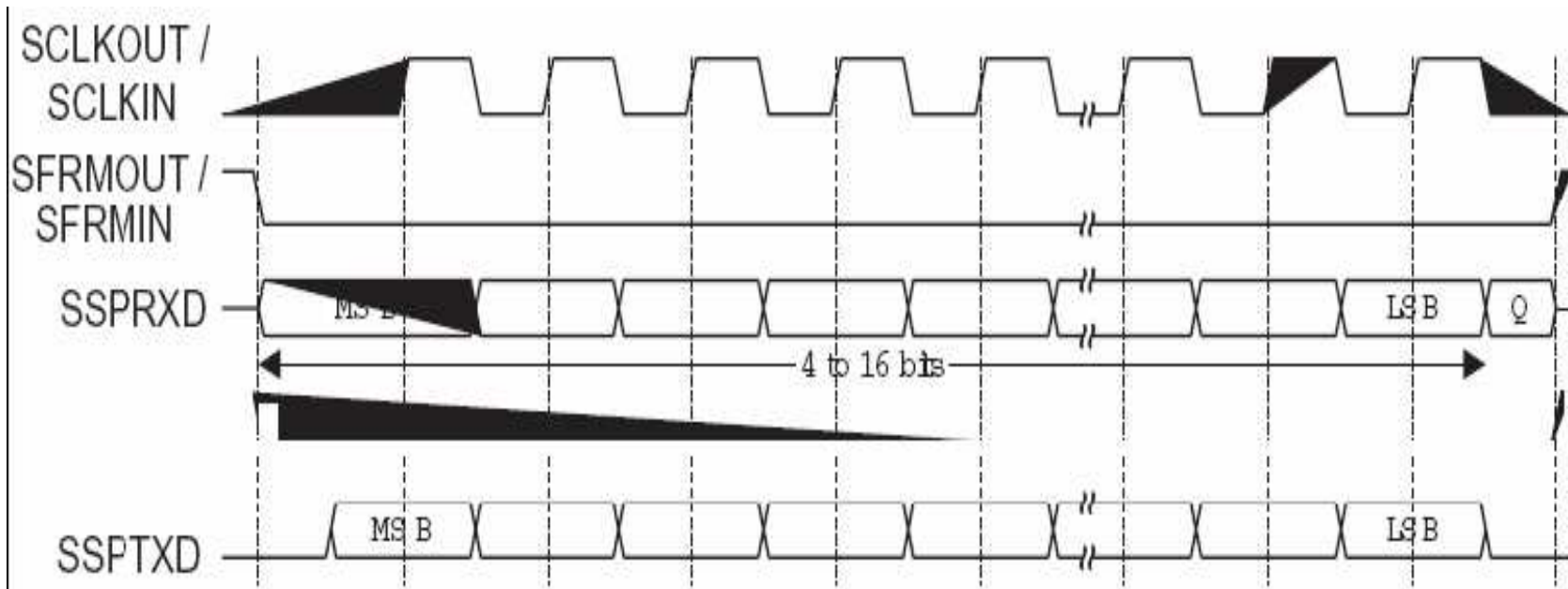
Pseudo code (TS7200 operation)

```
Initialize system
While true
{
  Get Data packet
  Output amount of data received
  Store Data packet
}
```

SPI Transmission Protocol

- q Developed by Motorola
- q Fast synchronous serial port communication
- q Master-slave architecture
- q DSP- master and TS7200-slave
- q 1 bit from master to slave n vice versa per second
- q Connections: clock, frame, ground, master-in-slave-out,
master-out-slave-in

SPI Transmission Protocol



Signal Diagram

Characteristics of SPI Transmission Protocol

- q Slave can transmit to Master only and only if Master is transmitting at the same time or else Slave has to wait
- q No acknowledgement sent either by Master or Slave
- q No guarantee of transmission quality
- q Can change polarity of signal
- q Additional bit can be induced for delay
- q Zero overhead

SPI Transmission Limitations

- q Transmission speed totally dependent upon Master's clock speed
- q If TS7200 would have been used as Master;
speed range – 29 kHz to 3.7 MHz
- q SPI – mainly intraboard protocol
- q When used interboard – causes EMI (Electromagnetic Interference)

TS7200 Limitations on SPI Transmission

- q Linux 2.4 kernel on TS7200 - not a RTOS (Real Time Operating System)
- q **Linux not a preemptive** – current task has to be finished; before starting new one
- q Hence OS can not respond immediately to event occurred as DSP
- q **Probability of data loss in consecutive samples**

TCP Implementation

- q To ensure, TS7200 receives data successfully from DSP
- q **DSP** implementation of TCP - **Master**
 - q data to slave in packets
 - q special 16-bit value
 - q acknowledgement from TS7200 is awaited
- q **TS7200** implementation of TCP – **Slave**
 - q count values received from DSP on SPI port
 - q one value – correct reception
 - q different value – incorrect reception

System Performance

- q Per minute - 20 seconds data is gathered, analyzed and stored
- q Power consumption is minimum
 - q approx. 7 watts – linear voltage regulator
 - q 4.5 watts – direct power supply, no regulator
 - q more efficient switching power supply needed
- q If TS7200 – directly mounted over DSP board – data transfer speed would improve

Conclusion

- q DSP not burdened with memory storage – merely processes data
- q Memory controller system – low power and economical
- q Successful data transmission from static storage to separate PC
- q DSP controls what data is logged
- q System could be – stand-alone data logger
- q Low cost, low power and miniature system