Radar Signal Processing and DSP Technologies for Automotive Applications

picture of building...

Table of Contents

1 Radar Systems Engineering .................................................................2
2 Narrow Band Radar Sensors and Controllers ........................................2
3 Signal Processing Techniques..............................................................4
   3.1 The Fast Fourier Transform ......................................................4
4 Radar: Range Resolution, Range ambiguity resolution ................................5
5 Radar: Measuring Angle of Arrival ..................................................5
   5.1 Mono-Pulse Technique ..............................................................5
   5.2 Phase Mono-Pulse Techniques ..................................................5
   5.3 Digital Beam-forming DBF ( for Multiple Channels ) ......................5
   5.4 Measuring a Signal Processing Algorithm ..................................6
   5.4.1 How silicon vendors measure and benchmark their products ..........6
5.5 Tools Mathworks ............................................................................7
   5.5.1 Matlab ....................................................................................7
   5.5.2 Matlab Simulink ......................................................................9
6 Radar: Target Tracking ....................................................................10
7 System Design Parameters and Considerations .........................11
   7.1 Applying New Technology .........................................................12
   7.1.1 Emerging Analog to Digital Converters ................................12
   7.1.2 Texas Instruments C2000 Floating Point DSC’s .........................12
   7.1.3 Texas Instruments C6000 Floating Point DSP’s ........................12
   7.2 Linear Math Libraries cblas_zgemm and zgemm ................................12
   7.3 Embedded Signal Processing, Esprit Algorithm for angle estimation ....13
   7.3.1 Forward Backward Smoothing ..............................................13
   7.3.2 Covariance Matrix ................................................................13
   7.3.3 Memory requirements ..........................................................13
   7.3.4 Embedded DSP Algorithms, Programming Languages and C++ ....14
8 References .......................................................................................14
   8.1 Radar References .......................................................................14
   8.1.1 Must Read References ..........................................................15
   8.1.2 SAAB Documentation ............................................................15
   8.1.3 AEACommon document repository .........................................15
   8.1.4 AEACommon1 document repository .........................................16
8.2 xPCDN .........................................................................................16
9 Appendix .........................................................................................16
   9.1 Appendix Long Range and Short Range Radar for Automotive Electronics 16
Automotive Radar and Sensor Systems is a mature and rapidly growing product market\(^1\) significant volumes of white papers and design descriptions are available as manufactures, compete for market share. In such mature markets software can often be the differentiator, when form factor hardware have been standardized by practical cost effective technologies. Cost constraints for automotive industry adoption of technologies significantly impacts system design component selection can critically effect design integration and effectiveness.

For automotive systems the signal processing real time constraint of 40ms Cycle time for digital signal processing is necessary beyond which the assumption that the speed of detected targets does not change is no longer valid. (Raw I/Q data, Offset compensation, detection, through Tracking). This constraint is related to both radar signal processing an the physical constraints of the automotive adaptive safety limits. Warning indications and driver reaction to applied braking 3G through automatically applied braking maximum 4G.

1 Radar Systems Engineering
predictive and effective system modeling.

2 Narrow Band Radar Sensors and Controllers

The narrow band devices include the Saab design as documented by Saab as well as the newer iteration, with new waveform processing. Digital Signal Processing Techniques differ in several ways:

Modulation Pulse Width vs Frequency Modulation (FMCW). Functional limitations in range resolution, field of view FOV.

For the goals of cost effectiveness the same RF and Digital hardware design has been used for both products. System design analysis suggests that two designs with such dramatic differences and limitations represent significant compromises in efficiency and power consumed.
Reviewing ARC4 Waveform2a Old Hardware...
Frequency plan?
Step sizes?
Assume FMCW saw tooth.? Min and max range...??
Aperture?  + or – 20 ???  FOV 50 + or – 25? 10m +60 degrees
The integration time extends beyond the propagation delay for blind spot
base propagation delay range/c

UWB Long Range Radar & Mid Range Radar large bandwidth (>500 MHz) pulsed transitions < 20%
- ACC (Adaptive Cruise Control) - Full Speed Range.
- Front, Pre-Crash Sensing / Collision warning.

Narrow Band 24GHz Short Range Radar, for blind spot and lane change assist applications.
Center Frequency 24.125GHz
Frequency range 24 – 24.25GHz
- Lane Change Assist. 60 meters (Freescale)
- Blind Spot Detection, BSD 20 meters.

Illustration 1: Short Range Radar - Blind Spot Detection Application with Lane Change Assist.
- Back Up Warning.
- Rear Cross Traffic Detection / RCTAssist.
- Side and Rear Pre-Crash Sensing / Collision warning.

Automotive Radar processing blocks
- Detection software  range velocity and angle
- Tracking software
- Warning algorithms
3 Signal Processing Techniques

3.1 The Fast Fourier Transform

DFT and IDFT

Discrete Fourier Transform and Inverse Discrete Fourier Transform, the Fourier transform pair.

\[ f(t) \leftrightarrow F(w) \]
\[ \tilde{X}(k) = \sum \tilde{x}(n)W^{kn/N} \]
\[ \tilde{x}(n) = \frac{1}{N} \sum \tilde{X}(k)W^{-kn/N} \]

The Fast Fourier Transform process converts time domain samples to frequency domain spectra, or
frequency domain. The FFT calculation of Discrete Fourier Transforms provides NlogN calculations an
improvement over the $N^2$ Disambiguation.

\[ \Theta \quad \sin(\theta) = \theta \quad \text{for all angles equal to zero.} \]

4 Radar: Range Resolution, Range ambiguity resolution

Minimum range separation of two objects traveling at the same speed that is necessary to detect 2
targets.

The time period between pulses transmitted at the carrier frequency is referred to as the PRT its
reciprocal is referred to as the pulse repetition frequency, PRF.

Multiple Pulse Repetition frequency PRF are used to differentiate targets who’s ranges are ambiguous.

If the distance is greater than $C/(2 \times \text{PRF})$ $C$ representing (3e8 speed of light and an approximate for
the prorogation of radio waves meters/second). A PRF of 3MHz would be necessary for range
resolution up to 50 meter. Cost effective hardware often can not meet this goal as a result range
ambiguity resolution processing is required.

A convolution technique
http://en.wikipedia.org/wiki/Range_ambiguity_resolution

5 Radar: Measuring Angle of Arrival

5.1 Mono-Pulse Technique:

“The angular position of a radar target is determined by the amplitude ratio of the received
signals in adjacent radar beams, typically referred to as monopulse technique, see Fig. 2-1 c).
To achieve a reasonable angular accuracy, a number of 2 to 5 mutually overlapping beams is
required to cover the azimuthal angular range of typical long or medium range applications.
The beam width depends on the aperture size of the antenna, which is limited by practical
reasons such as available mounting space at the bumper region.


5.2 Phase Mono-Pulse Techniques

5.3 Digital Beam-forming DBF (for Multiple Channels)

Digital Beam Forming requires multiple channels
For the Short Range Radar Narrow band Digital Beam Forming (DBF) follows the Detection algorithm,
beam forming returns range velocity and dbf-angle for the input sweep matrix with the most number of
detections...

Input: the single sweep matrix with the most number of detections provided by the detection algorithm
Output: An Array of range, velocity (from the input) and DBF angle for each of the N targets in the
input data.
For longer range forward looking radar DBF can provide error correction processing however it does not produce the angle resolutions needed and further processing in necessary. (ESPRIT or MUSIC)

5.4 Measuring a Signal Processing Algorithm

Without the ability to measure the relative com-putative size of a given signal processing algorithm, much programming, development and testing resources can result in a simple embedded CPU-bound processing problem. Fortunately silicon vendors provide ample information to help evaluate parts suitable to your application (DMIPS, FLOPS, cycles, …) Altera, Analog Devices, Xlinx, and Texas instruments.

Several algorithm implementation blocks have produced surprising execution results, I-Q Calibration, Detection(1.0, 2.0.. 3.0), Digital Beam-forming (versions 1,2,3) and Angle Estimation by 'Esprit'.

The Angle Estimation by Esprit algorithm historically 1986 known for its ‘computationally intensive’ methods in favor of accuracy assumed a single instruction cycle multiply only available to PC hardware in the estimation of embedded performance. No estimate was made of the considerable amount of data movement involved in the Esprit algorithm and the type of memory interface fabricated into the existing hardware. Once corrections were made the theoretical matrix multiply, EVD Eigen value decomposition, CSVD Complex Single Value Decomposition, and LLS linear least squares could be proportioned with historical FLOPS counts dating back to Fortran implementations. In the end it all made sense.

Version 1.0 of the Detection algorithm implementation, correctly accounted the well know N*log(N) FFT andIFFT in the FLOP count estimate, which unfortunately was not the highest calculation intensive operation when compared to the many discrete complex multiply operation, and median 256 point float calculations. For the Detection Algorithm profiling on PC hardware was effective in illustrating the many discrete multiplication of complex numbers far exceeded cpu cycles spent performing FFT and IFFT operations. Qsort a heavily optimized reference library function, incorrectly implemented by the TI DSP library and tools produced significant performance problems.

The Detection algorithm for version 2.0, significantly reduced the number of CPU cycles, along with a reduction in the number of antenna elements and sweep data samples. One fifth as many complex multiplications were used in the ‘pairing’ or disambiguation function. Execution times were directly proportional (53ms ~ 10ms) to this reduction in algorithm size.

Each step of the development process provides an opportunity to measure theoretical and expected algorithm performance. With cycle accurate CPU models for TI C6000 & C2000 families, as well as Analog Devices and Infineon, Mathworks and other tool vendors provide verification testing against estimation errors. If you are unable to afford licensed tools for measuring your algorithm several freely available open source techniques exist.

5.4.1 How silicon vendors measure and benchmark their products

Vendors are highly motivated in their competitive market place to fully explain how their product offerings compare against competing products, your design goals, and industry standard benchmarks such as MIPS and MFLOPS as well.

```c
start_timer
for ( x = 0; x <= 1Million; x++)
{
    cpu instruction (addition or multiplication)
}
end_timer
```
```
start_timer
for ( x = 0; x <= 1Million; x++)
{
    float * float;
}
end_timer
```

Other openly available benchmarks include Drystone Whetstone, Linpack_bench from Netlib.org....

Both Analog Devices (June 3rd) and Texas Instruments (July 19th) ported the C source code for the Detection 1.0 algorithm, and provided profile data, or number of instruction cycle counts, measured on actual CPU hardware. Detection 2.0 was again measured using profile tools for an indication of the effectiveness of the algorithm rewrite: RADAR_detection/doc/detection2.0_vs_1.0.ods Matlab provides a variety of tools to measure the effective number of CPU instruction cycles and FLOPS for various CPU families. TI C6000, C2000, ADI and Infineon..

5.5 Tools Mathworks

5.5.1 Matlab

Matlab the floating point mathematical analysis tool of choice, no longer supports and effective 'flops()' function and calculation. The remaining 'cputime()' functionality produces much confusion and significantly varying results in the Matlab profile tools. The matlab profile tools effectively abstract the necessary cpu cycle information and floating point operations from analysis. Few comparisons can be made between Matlab Profile reports and C source profile results successive profile sample produce alarmingly different results where the C profile tools converge on successive samples, reliable consistency. CPU specific plug-in modules available from Mathworks as well as openly available scripts can be used to automatically calculate, and model each CPU instruction cycle for any pipeline or execution unit.

Matlab results unfortunately do not necessarily lead to functioning and effective signal processing algorithms, too often a hardened Matlab algorithms overflows the processing boundaries of embedded sensor technologies available, at any cost.
### Profile Summary

Generated 28-Jun-2011 11:42:46 using cpu time

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Calls</th>
<th>Total Time</th>
<th>Self Time</th>
<th>Total Time Plot</th>
</tr>
</thead>
<tbody>
<tr>
<td>detection6_fastimplmentation</td>
<td>1</td>
<td>0.399 s</td>
<td>0.033 s</td>
<td></td>
</tr>
<tr>
<td>paimnet</td>
<td>4</td>
<td>0.144 s</td>
<td>0.086 s</td>
<td></td>
</tr>
<tr>
<td>EstAmbiaRanges3</td>
<td>4</td>
<td>0.054 s</td>
<td>0.000 s</td>
<td></td>
</tr>
<tr>
<td>dhf</td>
<td>9</td>
<td>0.063 s</td>
<td>0.063 s</td>
<td></td>
</tr>
<tr>
<td>FindMaxPeaks2</td>
<td>84</td>
<td>0.048 s</td>
<td>0.033 s</td>
<td></td>
</tr>
<tr>
<td>gen_target_returns</td>
<td>4</td>
<td>0.047 s</td>
<td>0.047 s</td>
<td></td>
</tr>
<tr>
<td>unscramble</td>
<td>224</td>
<td>0.033 s</td>
<td>0.033 s</td>
<td></td>
</tr>
<tr>
<td>ComputeChipAmpPhase3</td>
<td>4</td>
<td>0.032 s</td>
<td>0.032 s</td>
<td></td>
</tr>
<tr>
<td>median</td>
<td>328</td>
<td>0.031 s</td>
<td>0.000 s</td>
<td></td>
</tr>
<tr>
<td>median&gt;meanof</td>
<td>328</td>
<td>0.031 s</td>
<td>0.031 s</td>
<td></td>
</tr>
<tr>
<td>parabolic</td>
<td>381</td>
<td>0.016 s</td>
<td>0.016 s</td>
<td></td>
</tr>
<tr>
<td>fligir</td>
<td>4</td>
<td>0.016 s</td>
<td>0.016 s</td>
<td></td>
</tr>
<tr>
<td>close</td>
<td>1</td>
<td>0 s</td>
<td>0.000 s</td>
<td></td>
</tr>
<tr>
<td>close&gt;safegetchildren</td>
<td>1</td>
<td>0 s</td>
<td>0.000 s</td>
<td></td>
</tr>
<tr>
<td>close&gt;checkflags</td>
<td>1</td>
<td>0 s</td>
<td>0.000 s</td>
<td></td>
</tr>
<tr>
<td>close&gt;request_close</td>
<td>1</td>
<td>0 s</td>
<td>0.000 s</td>
<td></td>
</tr>
<tr>
<td>randperm</td>
<td>2</td>
<td>0 s</td>
<td>0.000 s</td>
<td></td>
</tr>
<tr>
<td>flags</td>
<td>2</td>
<td>0 s</td>
<td>0.000 s</td>
<td></td>
</tr>
<tr>
<td>hotstart</td>
<td>4</td>
<td>0 s</td>
<td>0.000 s</td>
<td></td>
</tr>
</tbody>
</table>
5.5.2 **Matlab Simulink**

A plan is needed for a method to make the essential measurement of algorithm size to effect suitable embedded devices selection.
5.6 Software Profiling

Software Profiling has been an effective means to measuring both the number of overall CPU cycles for an executing algorithm, and the number of floating point operations. Followed by measurements of execution times on evaluation or embedded hardware results often confirm expectations.

Analyzing Embedded Software:

- Octave: for confirmation of Matlab source scripts
- GNU profile tools for measuring relative execution times of C source code.
- Embedded software profile tools with cycle accurate simulation of embedded devices.
- Execution on embedded target hardware and validation against vendor benchmarks.

5.7 Processing Power

“Sizing up the processing power of one method over another depends on the type of data being processed and the algorithms doing the processing.”

Single Precision, Double Precision and Floating Point can greatly effect processing performance, further limited by the data resolution of 12 bit Analog to Digital Converters the type of data may be fixed if not SP. “Tesla class GPUs can peak at 1012 Floating Point Operations Per Second where Xilinx Vertex-6 are quoted at 150 GFLOPs.” Such technology thresholds could significantly effect the 1ms sweep goal resulting from workstation simulations.

Measurement of Floating point DSP performance presents unusual challenges. A theoretical calculation of CPU instructions without consideration of the actual number of pipeline instructions and Execution cycles (single precision 4, double precision 103) can lead to significant errors in estimated execution times. The problem is further complicated by variable optimization performance achieved though parallel instruction execution which can produce up to 20x performance improvement over Linear Predictive Coding. Unfortunately not all functions, operations, or algorithms can achieve fully parallelized execution for maximum optimized performance.

Fortunately DSP silicon vendors as well as DSP design solutions vendors have significant market incentives to measure and benchmark their performance against existing technologies. With system design experience it is possible to navigate the volumes of performance data to produce accurate embedded performance estimates which can be measured on embedded hardware. The Code Composer Studio Help menu “Tutorial Example List → Digital Signal Processors & ARM Microprocessors->TMS320C674x Low Power DSP->C674x Floating Point Benchmarks” references the same http://processors.wiki.ti.com/index.php/C674x_DSPLIB/c647x_dsplib_dev_notes.xls Benchmarks available in the DSP library resources.

Performance of TI DSPLIB resources are referenced for both OMAP-L137 and 674x implementations. A performance spreadsheet is provided with the docs folder of the C674x DSPLIB installation. (C:\CCStudio_v3.3\c674x\dsplib_v12\docs and http://10.106.10.119/pub/c674x_dsplib_dev_notes.xls). For the OMAP-L1x sample application performance cycle counts with IPC and without are provided on the web page for DSPF_sp_mat_mul(1.24ms), DSPF_sp_mat_mul_cplx (.812ms) and DSPF_sp_mat_trans(.721ms), of particular interest to Radar application processing using the Esprit algorithm.

---

2 http://www dsp-fpga com/articles/id/?4632 GE Intelligence Systems Peter Thompson
3 Reference 1. page 3-32 “Total Result Latency” MPYSP and MPYDP
4 Reference 2. E2E Forum C/C++ compiler group.
5 Reference 4 dsplib developers notes.xls DSPF_sp_mat_mul_cplx 553 Cycles (Absolute)
DSPLIB c674x/dsplib_v11 and dsplib_v12 have effective release dates of 6/25/2009 and 1/5/2010 respectively. Example source includes input data and benchmark results for several matrix math operations.

“C674x Floating Point Benchmarks Vector and Matrix” table formulas are provided for detailed calculation of cpu cycles. Benchmarks include instructions on how to download related source code C674x DSPLIB and C67x fastRTS packages.

Memory architectures must also be considered when determining expected embedded system performance. Embedded designs typically have much slower clock rates and narrower bus widths. Few simulation tools for embedded devices would provide a matrix for memory interfaces hence only relative CPU cycle counts would be useful. Target hardware currently supports 16 bits for 4Mbytes of memory half the bus width and ¼ the memory available to the reference design of the OMAP-L137.

The TMS320C28335 uses a Single access RAM (SARAM) 16 bit word memory interface the performance of this memory interface is expected to be specific to this architectures implementation. 256K 16bit word Flash (code) 32K 16bit word (critical code and data) SARAM and 256Kx 16bit ISSI Memory data as described in A24R-0043 “HW resources usage on F28335.....

Compiled esprit program executable binary with libraries 8.023805 Mbytes and 8.027934 Mbytes with smoothing conditional code ARC_5B 4Mbytes. Esprit program compiled against TI DSP libraries and BIOS currently 1Mbytes.

TI DSP Libraries and the SORT function, for reasons not sufficiently explained by the TI representative the sort function preforms poorly in both C6000 and C2000 embedded environments. Standard C library implementations (example glibc) outperform the TI library implementation with historically benchmarked and optimized implementations from statistics and mathematics computing groups.

6 Radar: Target Tracking

Tracking detected targets from radar sweep to radar sweep, while difficult and constrained by computational power, can compensate for noise, clutter and falsely detected targets. Weighted averages are compared against previous scans in real time, as a result tracking considerations and significantly impact the overall system design.

7 System Design Parameters and Considerations

Bandwidth difference between upper and lower frequencies, determines Narrow Band verses Ultra Wide Band sensors common in the Radar Sensor market. Resolution.

Angle of Arrival.

Important to remember is most embedded systems, especially those for automotive markets do not have access to the high speed memory systems and gobs of RAM common in todays PC hardware available to the mathematical algorithm development tools.

---

6 Difficult to understand why hand coded matrix multiply routines were being reviewed as late as August of 2010 with C++ types and double indexed arrays, long understood to block 'parallelization'.

Michael Nolin 11 of 20 September 15, 2011
Sample Interval, sample frequency.

12μs  83.333KHz   close to the 30KHz sampling rate of modern audio and professional audio
sampling rates
standard audio 20-20KHz human hearing and CD sampling rate of 44KHz..

7.1 Applying New Technology:

7.1.1 Emerging Analog to Digital Converters

"Today's high resolution ADC's top out around 200MillionSamples/sec new devices are expected with
250MS/sec performance." While current designs are dependent on 12 ADC's 14 Bit ADC's are
available as well as 16 bit ADC's in next generation devices 250MS/sec.

C2000 family and TMS320F28335 includes a 12bit ADC with 80ns conversion rate 12MHz sample rate

7.1.2 Texas Instruments C2000 Floating Point DSC's

Originally a product family of fixed point devices first introduced in the 1990's additions to the product
family have included additional floating point MAC's  2007 TMS320C28335. With its significant cost
advantages this family of Digital Signal Controllers, provides a variety of motor control features, PWM's,
ramp functions, and armature phase relations.

7.1.3 Texas Instruments C6000 Floating Point DSP's

The Texas Instruments C6xxx floating point processor family is a mature device family dating back to
1999-2000. While there have been improvements in clock rates (up to 1GHz) and multi core devices
these advances do not exist for the mid range and low power devices 5-4 watts. The C6747 device
represents a new integration of fixed point and floating point DSP features appearing in the past in
separate devices.

7.2 Linear Math Libraries cblas_zgemm and zgemm:

Esprit calls cblas_zgemm 6 times as well as getS1S2 function calls cblas_zgemm another 2 times for 8
calls to cblas_zgemm. These well defined linear algebra functions and API's made a logical starting
point for the embedded integration effort. The core mathematical processing of the Esprit algorithm is
handled by the 8 function calls to zgemm (complex) generic matrix multiplication functions.¹

\[ C := \alpha \text{op}(A) \text{op}(B) + \beta C \]

The repetitive nature of Linear algebra mathematics with matrices has been optimized over several
decades, optimized on the essential equation above.

---

¹ [Http://www.dsp-fpga.com/articles/id/?4457](http://www.dsp-fpga.com/articles/id/?4457) Curtiss-Wright Rob Hayecki
² Reference 9 B. General ESPRIT Algorithm 1-7
7.3 Embedded Signal Processing, Esprit Algorithm for angle estimation.

7.3.1 Forward Backward Smoothing

The practical application of smoothing involves the shifting or moving of data from a 12 x 64 matrix of complex values to a 9 x 256 resulting matrix. Effectively increasing the number of samples from 64 to 256 through a manipulation of statistical probability on a fixed data set. This becomes important because the number of samples compared to the number of considered variables effects the error rate and stability of the Estimation of the Covariance Matrix.

A method for joint angle-frequency estimation (JAFE) in the presence of coherent signals. “This method is based on constructing a forward-backward smooth spatio-temporal array data matrix by using the proposed forward-backward spatio-temporal smoothing method before performing the JAFE algorithm. The simulation results show that the proposed method is superior to the published spatio-temporal smoothing method. ISSN: 0013-5194 08 April 2003.

7.3.2 Covariance Matrix

“Estimating the Covariance Matrix” is the first computationally extensive operation of the Esprit Algorithm. To determine the Covariance Matrix a full data set of an M x N matrix is multiplied with the conjugate transpose of itself. See: wikipedia.org/wiki/Covariance_matrix and Estimation_of_covariance_matrix.

As derived from statistics it has applications using linear algebra and matrix mathematics. The method is not without its problems: as the “sample size is small in comparison to the number of considered variables is large (9 x 256)... estimators of covariance and correlation are very unstable.” Compare with: “Your results may vary.”

7.3.3 Memory requirements

Compiled esprit program executable binary with libraries 8.023805 Mbytes and 8.027934 Mbytes with smoothing conditional code ARC_5B 4Mbytes. Esprit program compiled against TI DSP libraries and BIOS currently 1Mbytes. PC_esprit implements a dynamic memory allocation memory model, this model cannot be implemented in the embedded environment as well as the execution cycles for dynamic memory allocation in an embedded environment would be an unnecessary waist.

The Esprit program cross compiled and linked using CCStudio v7.0.3 code generation tools for the c6747 uses. 892,815 bytes program 12,240 bytes data

DSPLINK 123,113 bytes
cmemk.ko 23406 bytes
dsplib_server 456378 bytes
dsplink.ko 1261756 bytes
lapack_LINUX.a and lapack_TIBIOS.a cblaswr 296,240 bytes code 1600 bytes data
libf2c.a 401990 bytes least squares and eigenvalue decomposition.
ulmage 1804720 bytes
zgemm and cgemm 60kbytes
7.3.4 Embedded DSP Algorithms, Programming Languages and C++

TI code generation tools and compiler limitations. A variety of programming languages are available in packaged libraries for mathematics and scientific calculations. For some this blurs the line between workstation computing, personal computers and embedded systems.

8 Radar System Data Flow

Data Flow analysis is an important System analysis tool.

Illustration 3: Signal Processing Data Flow Overview

Optimum Data Flow analysis that yield .. is a feed back into the system design process.. into which highly detailed descriptions that can only be obtained from the embedded source decomposition. The Data Flow design may itself go through several iteration phases (example: MRR). As new silicon technologies and SoC devices become available designs must be updated to take full advantage of continuing hardware integration.

The MRR effort provides an important case study to the detailed data flow analysis, with subsequent cache – internal RAM balancing for the multi layered internal memory, producing a significant
performance boost. New Data Flow diagrams were needed to reflect the new system.

The Narrow Band radar devices as originally defined by Saab Software Design Document have not been through the first phase of Data Flow analysis, the added performance many not be necessary as the product achieve its design goals.

9 References:

1. Texas Instruments TMS320C6000 Optimization Workshop Student Guide
2. TI E2E Community “Use of C++ <complex> types and measured performance” TI C/C++ Compiler Forum Clear Quest SDOWP ID#SDSCM00037600 Georgem.
   c647x_dsplib_dev_notes.xls
   TI DSPLIB “Legacy ASM Implementation from C67x” DSPF_sp_mat_mul_cplx.asm
6. www.wiki.ti.DPS64x
7. netlib.org cblas_zgemm
10. ESPRIT Beam Forming for the Autoliv Long Range Radar, Bruce Labitt.
11. “Singular Value Decomposition – A Primer” Sonia Leach Department of Computer Science Brown University Providence RI 02912. DRAFT VERSION. (Postscript) Ghost view (1994)
13. Unitary ESPRIT: How to Obtain Increased Estimation Accuracy with a Reduced Computational Burden Martin Haardt, Student Member, IEEE and Josef A. Nossek, Fellow IEEE, IEEE Transactions on Signal Processing, Vol 43 No. 5, May 1995

9.1 Radar References:

2. “Radar Systems for Planetary Exploration” Mike Taylor taylor_michael_a5@cat.com
   S:\AEL\Common\Trace\cbuProjects\Engtest\ACC\Presentation.ppt
3. “Automakers prepare for multicore processors” aei-online.org (automotive Engineering International Online) Kevin Jost 2007 S:\AEL\Common\Trace\cbuProjects\Engtest\ACC
5. Honda SPEC, BSI UNIT 10/22/2010 S:\AEACommon\projects\Blind_Spot_Radars\Honda BSI
   “Automotive Radar Millimeter wave technology
9.1.1 Must Read References.

Openly available references cross referenced and reviewed by many contributors.


9.1.2 SAAB Documentation

Saab References, SRR-NB Radar and Blind Spot: S:\AEACommon1\active_safety\24GHz\Saab Transfer Documents

A24R-0019 Automotive Development Environment 3-12-2009
- a dozen different tools used to develop the software, pictures and instructions for how to install VS2005.
A24R-0025 Radar Sequencer unit, SW Implementation details 1/30/2008
- how to program flash...
A24R-0043 (v1.0C) HW resources usage on F28335... (and other obsolete parts).
  - “radar sensor block diagram”
  - memory usage
  -
  S:\AEACommon1\active_safety\24GHz\Saab Transfer Documents

S:\AEACommon\projects\Radar_24\ARC_4B\ARC4B_Docs

9.1.3 AEACommon document repository

1. S:\AEACommon\projects\Radar_24\ARC_4B\ARC4B_Docs
2. S:\AEACommon\active_safety\24Ghz\Radar Learn\Stuff from Larry Laryy stopczynski 01/09/09
3. NB Radar Senor Block Diagrams.

9.1.4 AEACommon1 document repository

1. S:\AEACommon1\active_safety\24Ghz\Radar Learn\Fundamentals of Radar Signal Processing... 2007
2. www.pe.gatech.edu Georgia Institute of Technology
3. S:\AEACommon1\active_safety\24Ghz\Radar Learn\Skolnik Introduction to Radar Systems Third Edition
5. S:\AEACommon1\active_safety\24GHz\Saab Transfer Documents

9.2 xPCDN

S:\AEACommon1\active_safety\24GHz\NB_BSD_2011 pcdn_sw.doc Saab Bofors Dynamics AB.. 2002-12-17
XPCDN SW documentation  xPCDN_Aug16_after_Japan

Makefile using a gcc compiler to compile Sensor Files into dll's  wow!! target to create dependency needs
cygwin and gcc of course to run!!

xpcdn a Matlab program argc argv  command interpreter and GUI interface..  yeesh.  Total relic of legacy
stuff.

10 Appendix

netlib.org CLAPACK-3.2.1,
netlib.org LAPACK-3.2.2
netlib.org ATLAS
numbpy sourceforge.net.

10.1 Appendix Long Range and Short Range Radar for Automotive Electronics

The competitive market place for long range radar includes several vendors with many design
approaches to consider. The majority include 77-80GHz devices with an impressive low power
performance of 7W. Few vendors are promoting 24GHz as devices operating at this frequency are in-
elligable for new product introduction in EU as well as not permitted as soon as 2013.

26GHz 'temporary' band as it cannot be introduced in EU or Japan for interference with radio
frequencies. See Automotive Radar – Status and Trends, Martin Schneider 2005.

Continental Temic (ADC in cooperation with M/A-Com) 'ContiTemic'
TRW Automotive (Autocruise)
Robert Bosch Automotive
Delphi ACCR ASIC PPC
Denso Automotive 77GHz 150Meters  http://www.globaldenso.com
smartmico.de 24Ghz radar production 2006 narrow band radar multiple sensors supplier to Hella
Velego-Raytheon Commercial group Ford automotive www valeo.com inventor of MMIC
Hella Volks Wagon Audi  www.hella.com 24 Ghz UWB for short range applications and a narrow-
band FMCW radar in teh license free 24 Ghz ISM band for a maximum range for 70m
RoadEye
Fujitsu Ten
Hitachi laser diode and radar.
Siemens VDO Video Sensor

Other Radar Applications
http://www.navtechradar.com. Airport, Security Systems, Managed Motorways...

Providing silicon solutions specific to automotive radar applications. SoC devices.
  TriQuint Semiconductor
  Freescale
  Analog Devices
  Infineon (formally Siemens Semiconductor)

10.1.1 Terms and Abbreviations
ACC Adaptive Cruise Control
BiCMOS, 77GHz silicon solutions for Freescale/ Astyx
BSI Blind Spot Indicator, (see Mercedes M class youtube trailer)
BSD Blind Spot Detector.
FCW Forward Collision Warning.
FMCW Frequency modulated continuous wave.
FOV Field of View radar and optics terminology usually expressed as an angle across a centerline
GaAs Gallium Arsenide semiconductor process common to MMIC technologies.
MMIC Monolithic Microwave Integrated Circuit examples Raytheon, TriQuint, STMicro semiconductor.
ePWM enhanced Pulse Width Modulation available on the TMS320C28335
PWM Pulse Width Modulation.
SAR Synthetic Aperture Radar
Sensor Fusion combining multiple sensor technologies, camera + radar, laser + radar, thermal + laser.
SiGa Silicon Germanium Freescale...

Technologies emerging in Automotive Electronics and Radar:
  • ASIC signal processing
  • multiple forward looking antenna’s 2 and as many as 3 smartmicro
  • FMCW Frequency Modulation Continuous Wave ARS300 Doppler’s principle. One cycle measure.
  • Multi mode radar long and short range in a single sensor (Delphi)

Radar Sensing and Communications for Automotive applications.
Important Student contributions to radar technologies, a practical exercise on building and collecting data from a SAR radar.

Northeastern University College of Engineering.

- http://www1.coe.neu.edu/~eric_w/RadarIAP.html

10.2 Bandwidth

The bandwidth of a system “consistency of the magnitude” in a system. The interval of positive frequencies over which the magnitude $|H(w)|$ remains within a given numerical factor -3dB half power. $1/\sqrt{2}$ in voltage or $1/2$ in power.

\[ |H_1(\omega)| \]

\[ \omega_1 \]

Radians per second
10.2.1 Convolution
Convolution in the time domain corresponds to multiplication in the frequency domain.

10.3