

November 2008

Introducing Industry's Highest Performing Programmable DSP Enabling OEMs Development and Deployment of Next-Generation LTE Broadband Wireless Base Stations

PN121

Barry Stern Multicore DSP, WW Marketing Manager







- Emerging Broadband Wireless Standards
 - Timelines, Throughputs, and Challenges
- MSC8156 Multicore DSP
 - Block Diagram
 - Next Generation StarCore® DSP core, SC3850
 - · Features and Benefits
- MSC8156 Baseband Solution
 - Target Applications
 - Development Tools
- Multicore DSP Roadmap
- Summary of Benefits
- Rules of engagement and product timelines



3G Evolution – from Thin to Thick Data Pipe

3G-LTE Significantly Outperforms 3G Standards

Increasing integration for high data rates and low latencies

WCDMA

0.5 Mbps

at 5MHz

Algorithm differentiation and flexibility require high-performance multicore DSP for programmability combined with integrated baseband accelerators for cost and power efficiency







Broadband Wireless Technology Timelines





Freescale Solutions Meets 3G-LTE Challenges







MSC8156 Multicore DSP



Freescale ™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2008.

MSC8156 Highlights

- Target Wireless Basestation Systems
 3G-LTE, TDD-LTE, WiMAX, HSPA+ and
 - TD-SCDMA
 - Meets all leading future wireless technologies

Single MSC8156 Handling

- Single Sector 3G-LTE, TDD-LTE
- Multi-sector WIMAX
- Multi-carrier TD-SCDMA
- Multi-sector HSPA along with external chip rate acceleration

Pioneering Expertise

- Leveraging multicore architecture expertise by introducing its 4th generation multicore DSP
- First to implement Turbo/Viterbi accelerators in DSP—compliant to latest OFDMA standards

Highly Efficient Memory Hierarchy

- Large on die low latency memory:
 - 6x512KB of L2/M2 + 1MB M3 = 4Mbyte

MAPLE-B Accelerator

Highest throughput, multi-standards compliant, re-programmable



MSC8156 Device Performance

- •Optimized Programmable Performance
 - •Based on New generation SC3850 DSP core
 - •Delivers up to 48GMACS
 - and 6GHz effective
 - performance
 - •Added Intelligent application specific accelerators
- •High-speed standard interfaces

•2xsRIO, 2xSGMII, PCI-Express

- •Highly optimized multilevel memory
- •High speed DDR3 interfaces



MSC8156/E – Broadband Wireless DSP

6x SC3850 Cores Subsystems (6GHz/48GMACS)

- New SC3850 DSP core at up to 1GHz (1GMACs 16b x 16b)
 - 512 Kbyte unified L2 cache or M2 memory.
 - 32 Kbyte I-cache, 32Kbyte D-cache, WBB, WTB, MMU, PIC
 - Fully Programmable
- Internal/External Memories/Caches
 - 1056 KByte M3 shared memory (SRAM)
 - Dual DDR3/2 64-bit interfaces at 800 MHz
- CLASS Chip-Level Arbitration & Switching Fabric
 - Non-Blocking, fully pipelined, low latency
- MAPLE-B Baseband Accelerator
 - Turbo/Viterbi Decoders up to 200/115 Mbps
 - FFT/DFT accelerators up to 280 Msps FFT/175 Msps DFT
 - Standards: 3G-LTE, 802.16, WCDMA, CDMA2K
- Security Engine
 - AES, Kasumi, SNOW-3G, SHA, RC-4 (in MSC8156E)
- High speed Interconnects
 - Dual 4x/1x Serial RapidIO at 1.25/2.5/3.125 Gbaud
 - PCI-e 4x/1x
- Ethernet
 - Dual Gigabit Ethernet ports (SGMII)
- TDM Highway
 - 240Mbps, divided into 4 ports of 8x E1 each.
- DMA Engine 32 ch memory-to-memory
- Other Peripheral Interfaces SPI, UART, I2C, 32 GPIO, 16 Timers, 96KB boot ROM, JTAG



Technology

45nm, 1V core, 2.5, 1.8/1.5V I/O FCBPGA, 29mmx29mm, 1mm pitch, RoHS



The New SC3850 Core Subsystem



- SC3850 Core DSP
- MMU
- L1 Instruction & Data cache
- Unified L2/M2 Cache
- Debugging unit
- PIC
- Timers



New SC3850 DSP Core improvements vs. SC3400

Feature	Code Type	Highlight	Benefit
-Dual Multiply -FFT support	DSP	-Eight 16x16 multiplications per cycle -Complex operations -Mixed/Double precision MPY support -FFT dedicated ISA	-Double the throughput of convolution based kernels, complex arithmetic and mixed/double precision multiplications -Significant FFT cycle reduction with SNR increase
-Control handling ISA	CTRL	-Parallel condition computation -Pointer-Relative Accesses -AGU logic instructions -Immediate DALU register TFR -NOBTB instruction	-Accelerate decision making in control code -Faster accesses to C structures -Improved compiler support (speed and size)
-Improved µ-Arch	DSP/Ctrl	-Improved HW loop operation -Deeper speculation depth -BTB enlargement -Deeper RAS stack	-Significant control cycle improvements and a friendlier compiler target -Achieving better performance at smaller code size
-Cache support instructions	DSP/Ctrl	-DFLUSH/DSYNC, D/PFetch, DMALLOC, SWPFETCH and more -For both L1 and L2 caches	-Improved cache operations



MSC8156/MAPLE-B Throughput & Compliance Data

Technology	Accel.	Standard Compliance	Data Rates	Comments	
3G-LTE, TDD-LTE	Turbo	3G-LTE (Evolved UTRA) turbo decoding as specified in 3GPP TS 36.212, section 5.1.2.2	up to 160 Mbps (8 iterations) up to 200 Mbps (6 iterations)	Max Log Map or Linear Log Map (MAX*) Support Rate-De-Matching (sub-block de-interleaving and de-interlacing) CRC calculation	
	Viterbi	3G-LTE (Evolved UTRA) channel decoding as specified in 3GPP TS 36.212, section 5.1.2.1	up to 100 Mbps (K=7 with tail biting)	Multi-iteration decoding	
	FFT/DFT	FFT sizes - 128, 256, 512, 1024, 2048 points DFT sizes - Variable lengths DFT/IDFT processing of the form 2 ^k ·3 ^m ·5 ⁿ ·12, up to 1536 points	FFT – up to 280 Mega samples/sec DFT – up to 175 Mega samples/sec	Advanced scaling options Guard bands insertion in iFFT	
	CRC	Transport and Code Block CRC for UL and DL	up to 12 Gbps	CRC check or insertion	
WiMAX	Turbo	WiMAX OFDMA turbo decoding as specified in IEEE® 802.16 [™] -2004 standard	up to 156 Mbps (8 iterations) up to 195 Mbps (6 iterations)	Max Log Map or Linear Log Map (MAX*) Support Rate-De-Matching (sub-block de-interleaving and de-interlacing)	
	Viterbi	WiMAX OFDMA turbo decoding as specified in IEEE® 802.16 [™] -2004 standard	up to 100 Mbps (K=7 with tail biting)	Multi-iteration decoding	
	FFT	FFT sizes - 128, 256, 512, 1024, 2048 points	FFT2048 – up to 280 Mega samples/sec FFT1024 – up to 350 Mega samples/se	Advanced scaling options Guard bands insertion in iFFT	
	CRC	PHY Burst CRC for UL and DL	up to 12 Gbps	CRC check or insertion	
HSPA+	Turbo	3GPP turbo decoding as specified in 3GPP TS 25.212, section 4.2.3.2.	up to 131 Mbps (8 iterations) up to 165 Mbps (6 iterations)	Max Log Map or Linear Log Map (MAX*) Support EDCH Rate De-Matching	
	Viterbi	3GPP viterbi decoding as specified in 3GPP TS 25.212, section 4.2.3.1.	up to 115 Mbps (K=9 zero tail)		
Programming Model	ALL	Buffer descriptors paradigm for allocation of data and control parameters Sharing of MAPLE-B modules in multiple devices using SRIO 'GO' command activation, no DSP core pre-processing or intervention are required			



6GHZ Performance Leapfrogs Previous Generation DSPs



- ► 6GHz equivalent performance, 1GHz a core
- ► New SC3850 DSP core improves signal processing & control code performance
- Integration of high throughput and standard compliant baseband accelerators
- ► High throughput external interfaces and highly efficient memory hierarchy
- 45nm most advanced process technology



Highest Performance, Multistandard DSP for Basestations

Multistandard Baseband Solution





- Processing power equivalent to 6GHz DSP plus FPGA
- Based on next generation SC3850 programmable DSP core
- Embedded with high throughput multistandard baseband accelerators
- Industry's first 45nm, six core DSP ideal for infrastructure
 Ready for deployment of cost-optimized basestations of 3G-LTE

* Requires an external Chip rate ASIC



MSC8156 Value Proposition

Attributes	Features	OEM Benefits
High Performance	 Industry's highest performance programmable DSP, 6 GHz raw performance and 48GMAC 16x16 operations Manufactured on 45nm industry's most advanced process technology MAPLE-B baseband accelerator supports throughput required for 20MHz single sector 3G-LTE High speed interconnects dual sRIO, dual GbE and PCI-Ex High speed DDR3 – dual 64b 800MHz Highly efficient memory hierarchy with 4Mbyte of Internal memory 	 Much higher densities and throughputs than previous gen. Effectively handles latest wireless standards requiring high speed and low latency throughput Cost optimized Enables freedom of scalability from 5MHz to 20MHz Single MSC8156 handles 3G-LTE 20MHz with MiMO or WiMAX 10MHz Multi sector Efficiently handles 3G-LTE data rates
Programmability	 MSC8156 – fully programmable DSP cores MAPLE-B accelerators supports latest wireless standards and consists of pre-configured and re-programmable RISC engines 	 Enables integrating OEM's intellectual property using highly efficient compiler and tools Enables designing of unified, upgradable, standard compliant and multistandard basestations
Efficiency	 FPGA-less PHY solution, development of FPGA or ASIC is not required Ultra-efficient DSP specifically designed to handle the computationally intensive processing of baseband requirements of LTE data bandwidth Industry standard high speed interfaces Low power solution 	 Expedite time to market and reduce development costs Significant lower cost channel card Reduced development effort, power consumption and cost-effectiveness play key factors in basestation design Enable daisy-chaining of sRIO eliminating the need for sRIO switch Efficient board routing
Upgradability and Compatibility	 Safe roadmap continuum for new DSP cores, DSP devices and process technologies "C" base language portable and upgradable solutions from existing products Binary code compatibility from one generation to another 	 Freedom to change from legacy silicon vendors Enable seamless transition to next generation devices Provides the opportunity to take full advantage of LTE capabilities today





Ideal for 3G-LTE & WiMAX Basestations





Freescale [™] and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2008.

MSC8156 in 3G-LTE & WiMAX Processing Chain



- Layer 1 PHY, Layer 2 MAC
- No FPGA/ASIC required
- Scalable platform



3G-LTE, 20 MHz FDD, 3 Sectors – Layer 1 Architecture



Scalable Solution

- Covers configurations up to 20 MHz, 430Mbps downlink & 230Mbps uplink,1 carrier/3 sectors, with 4Tx and 4Rx antenna
- Each MSC8156 DSP covers PHY handling uplink and downlink for 1 sector
- Separate Serial RapidIO® connection for both data and control paths



WiMAX 802.16e, 10 MHz TDD, 3 Sector – Layer 1 Architecture



Scalable Solution

- Covers configurations up to 10 MHz, 300Mbps downlink & 42Mbps uplink,1 carrier with 3 sectors, with 4Tx and 8Rx antenna and beamforming
- Allows for direct connectivity to the backplane and antenna without a switch by daisy-chaining Serial RapidIO®



MSC8156 Easy to Use Development Tools – CodeWarrior 10.x

- CodeWarrior® IDE
 - Eclipse-based



- StarCore® Build Tools
 - 'C' & 'C++' Optimizing Compliers, Linker, ASM, Utilities
- Debugger
 - Multicore and Multi-DSP support
 - Full access and control
 - USB and Ethernet TAP probes for silicon debug
- Trace & Profile
 - Support of advanced debug & profiling capabilities/analysis
 - MSC8156 silicon & simulator targets

Software Simulators

- Core Platform Cycle Accurate
- Device Functional Accurate

SmartDSP-OS RTOS

- Field deployed
- Fully pre-emptive
- Royalty free
- Built-in device drivers for MAPLE-B, Serial RapidIO, Eth, TDM, DMA, SPI, I2C

Exacts Message Type Cover Banch C	Detailer Unknown Ad Unknown Ad Address Det Address Det	<pre>Perfect(); Perfect(); Perfect(); Perfect(); Perfect(); Perfect(); perfec</pre>	Adds. Instruction 0344 wore, 1 (gp-ax), d5 0356 jpax/b wore, 1 (gp-ax), d5 036 add 431, d5 030 add 431, d5 040 pop, 4320, c5 040 pop, 4320, c5 040 pop, 7 0406 pop, 75 0407 pot, 75 0408 pop, 75 0408 pog, 75 0408 pog, 75 0408 pog, 75 0408 pog, 75 0418 pog, 75 0418 pog, 710 0418 pog, 710 0418 pog, 710			
2 Direct Starch 3	Ukinovan Ad Ukinovan Ad Ukinovan Ad Ukinovan Ad Ukinovan Ad Ukinovan Ad Ukinovan Ad Ukinovan Ad Ukinovan Ad Address Oat Address Oat	<pre>Perfect(); Perfect(); Perfect(); Perfect(); perfec</pre>	014 sove 1 44.(r4) 026 ipsarbb nove 1 (sp-8),d5 036 a dysarbb 036 a dysarbb 036 a dysarbb 036 a dysarbb 046 subs 76,sp 0464 subs 76,sp 0464 subs 76,sp 0466 FuncEad_Perfect 0400 funcEad_Pe			
2 Direct Barch 3 Direct	Usitnoven Ad Usitnoven Ad Usitnoven Ad Usitnoven Ad Usitnoven Ad Usitnoven Ad Usitnoven Ad Usitnoven Ad Address Out Address Out	<pre>Perffest(); Perffest(); potcaleep(10); } } void Perffest(void) { volatile int index; int nCount = 100; volatile int numberArray[PERFNDM]; volatile int tempErray[PERFNDM]; index = 0;</pre>	0 056 ipmarkh more i (sp-88) cell 0 056 add #11, d5 0 056 nove v #022, r6 0 040 nov v #022, r6 0 040 nov v #022, r6 0 040 nov p f6 0 040 pop r6 pop r7 0 040 FuncEnd PerfTest 0 040 cev v #0448, r6 pop r6 0 040 nov v #000 nov v #0448, r6 pop r6 0 040 nov v #0488, r6 0 040 nov v #04888, r6 0 040 nov v #048888, r6 0 040 nov v #0488888, r6 0 040 nov v #0488888, r6 0 040 nov v #04888888, r6 0 040 nov v #04888888888888888888888888888888888888			
2 Direct Barch 3 Direct	Unknown Ad Unknown Ad Unknown Ad Unknown Ad Unknown Ad Unknown Ad Unknown Ad Unknown Ad Address Dei Address Dei	<pre>Perfect(); Perfect(); gotosleep(10); } ; void Perfect(void) { volatile int index; int nCont = 100; volatile int numberary(PERFNUM); volatile int numberary(PERFNUM); index = 0;</pre>	norme 1 (sp-8), d5 03f es d4 41, d5 03f es d41, d5 03f es norme. 1 d5, (sp-88) 03f es norme. 1 d5, (sp-88) 0404 subs 76, sp 0404 subs 76, sp 0406 pp r5 0406 FuncEnd_PerfTest1 0410 - PerfTest2 sub d0, d0, d0 push r6 push r6 041 sc 04 for w #5648, r6 nop 04 dest 1 fc 04 dest 1 fc			
2 Direct Barch 3 Direct	Usinovan Ad Usinovan Ad Usinovan Ad Uninovan Ad Uninovan Ad Address Out Address Out	<pre>void Perffest(); gotoaleep(10); } } void Perffest((void) { volatile int index; int nCount = 100; volatile int temperay[PERFNDM(; volatile int temperay[PERFNDM(; index = 0;</pre>	03fa add #11.d5 03fa nova u.d5.(sp-04) 04fa nova u.d5.(sp-04) 04fa nova u.d5.(sp-04) 04fa nova f.d5 nova 04fa nova f.d5 nova 04fa nova f.d5 nova 04fa funcEad PerfTest 1 04fa funcEad PerfTest 1 04fa funcEad PerfTest 2 04fa funcEad PerfTest 2 04fa funcEad PerfTest 2 04fa func v #1648.r6 04fa down v #1648.r6 04fa down v #1648.r6			
2 Direct Barch 3 Direct	Unknown Ad Unknown Ad Unknown Ad Unknown Ad Address Dot Address Dot	<pre>verifiests(); gotoaleep(10); } } void PerfTest1 (void) { volatile int index; int nCont = 100; volatile int numberArray(PERFNDM); volatile int tempArray(PERFNDM); index = 0;</pre>	010 more 1 dS (mp-80 011 more # \$520 r6 0404 subs 76 mp 0404 subs 76 mp 0406 pop r6 0406 pop r6 0406 pop r6 0406 pop r7 0400 po			
2 Direct Barch 3 Direct	Unknown Ad Unknown Ad Unknown Ad Unknown Ad Address: Du Address: Du	<pre>gotcalep(10); } void Perffest(void) (volatile int index; int nConst = 100; volatile int numberArray[PERFNDM]; volatile int tempArray[PERFNDM]; index = 0;</pre>	037 store v #328 rd 040 roop f 040 room f 04			
2 Direct Barech 3 Direct Barech 4 Direct 4 Direct 4 Direct 4 Direct 4 Direct	Unknown Ad Unknown Ad Address: Dol Address: Dol Addres	<pre> void PerfTest1 (void) volatile int index; int nConst = 100; volatile int numberArray[PERFNDM]; volatile int tempArray[PERFNDM]; index = 0; </pre>	9402 scp sp 9404 suba *6 sp sp 9406 suba *6 sp sp 9407 suba *6 sp sp 9408 pop *6 sp 9409 pop *7 sd 9409 pup *6 sp 9409 pup *7 sd 9410 pup *16 sp 9410 pup *16 sd 9410 pup *16 sd 9411 pup *16 sd 9412 pup *16 sd 9413 pup *16 sd 9414 pup *16 sd 9415 pup *16 sd 9416 pup *16 sd 9417 pup *16 sd 9418 pup *2 sd 9419 pup *3 sd<			
Index Brack with ap Index Brack with ap Index Brack Index Mrine Index Mrine Index Mrine Index Mrine Index Brack Index Mrine	nc Unknown Ad Address: Ori Address: Ori	<pre>void Perflest1 (void) { volatile int index; int nConst = 100; volatile int numberArray[PERFNUM]; volatile int tempArray[PERFNUM]; index = 0;</pre>	0404 suba r6.sp 0406 pp r6 pp r7 0406 rts 0406 Funchd_PerfTest 410 _PerfTest2 9418 nove v #5440.r6 0418 nove v #5440.r6 0418 nove v #5440.r6			
10000 (bloch white) 10000 (bloch white) 2 Direct Stanch 2 Direct Stanch 2 Direct Stanch 2 Date Stanch 2 Date Read 2 Date Write 2 Date Read 2 Date Write 2 Date Read	Address: Oxi Address: Oxi	<pre>void PerfTest1 (void) { volatile int index; int nConst = 100; volatile int numberArray[PERFNDM]; volatile int tempArray[PERFNDM]; index = 0;</pre>	0406 pop r5 0406 pop r5 0406 pta 0406 pta 0400 pta			
2 Direct Blanch 2 Direct Blanch 2 Direct Blanch 2 Direct Blanch 2 Data Read 2 Data Read 2 Data Read 2 Data Read 2 Data Read 2 Direct Blanch 2 Direct Blanch 2 Direct Blanch 2 Direct Blanch 2 Direct Blanch	Address: Oxt Address: Oxt	<pre>void PerfTest1 (void) { volatile int index; int nCount = 100; volatile int numberArray[PERFNUM]; volatile int tempArray[PERFNUM]; index = 0;</pre>	0406 ppp r7 040c rts 040c rts 040c rts 040 penzna verfrest sub d0.d0.d0 push r6 push r7 0418 nove w #5648.r6 nop doen3 #Sc8 022 denump2 (1109.0)			
Direct Branch Direct Branch Direct Branch Date Read Date Read Date Read Date Read Date Read Date Write Direct Branch Direct Branch Direct Branch Direct Branch Direct Branch Date Read	Address: Oxt Address: Oxt	<pre>void Perffect (void) { volatile int index; int nCount = 100; volatile int numberArray[PERFNDM]; volatile int tempArray[PERFNDM]; index = 0;</pre>	 040c rtmcEnd_PerfTest: 0410 FuncEnd_PerfTest: 0410 JPerfTest2 sub d0.d0.d0 push r5 0418 nove. v #5648.r6 nop dcen1 #Sc8 0424 desture2 (1199.0) 			
2 Direct Branch 2 Direct Branch 2 Data Read 2 Data Read 2 Data Write 2 Data Read 2 Data Write 2 Data Write 2 Direct Branch 2 Direct Branch 2 Data Read 2 Data Write 2 Data Read 2 Data Read 2 Data Read 2 Data Write 2 Data Writ	Address: Oxt Address: Oxt Address: Oxt Address: Oxt Address: Oxt Address: Oxt Address: Oxt Address: Oxt Address: Oxt Address: Oxt	<pre>volatile int index; int nCont = 100; volatile int numberArray[PERFNUM]; volatile int tempArray[PERFNUM]; index = 0;</pre>	0400 FuncEnd_PerfTest: 0410 _PerfTest2 sub d0,d0,d0 push r6 push r7 0418 nove v #548.r6 nop doen3 #Sc8			
2 Data Read 2 Data Write 2 Data Write 2 Data Write 2 Data Write 2 Data Write 2 Direct Branch 2 Direct Branch 2 Data Write 2 Data Write 2 Data Write	Address: DxC Address: DxC Address: DxC Address: DxC Address: DxC Address: DxC Address: DxC Address: DxC Address: DxC	<pre>volatile int index; int nCount = 100; volatile int numberArray[PERFNOM]; volatile int tempArray[PERFNOM]; index = 0;</pre>	0410 _PerfTest2 sub d0, d0, d0 push r6 push r7 0418 nove.v #\$648, r6 nop doem1 #\$c8			
2 Data Write 2 Data Write 2 Data Read 2 Data Write 2 Data Write 2 Direct Branch 2 Direct Branch 2 Direct Branch 2 Direct Branch 2 Data Write	Address: DxC Address: DxC Address: DxC Address: DxC Address: DxC Address: DxC Address: DxC Address: DxC	<pre>volatile int index; int nCont = 100; volatile int numberArray[FERFNUM]; volatile int tempArray[FERFNUM]; index = 0;</pre>	sub d0,d0,d0 push r6 push r7 0418 nove v \$\$648,r6 nop deen3 \$\$c8 0244 deenture (7109.0)			
2 Data Read 2 Data Read 2 Data Read 2 Data Write 2 Data Write 2 Data Write 2 Direct Branch 2 Direct Branch 2 Data Read 2 Data Write	Address: DxC Address: DxC Address: DxC Address: DxC Address: DxC Address: DxC Address: DxC Address: DxC	<pre>int nCount = 100; volatile int numberArray[PERFNUM]; volatile int tempArray[PERFNUM]; index = 0;</pre>	Dush r6 push r6 0418 nove.v #\$648.r6 nop doen3 #\$c8			
2 Data Write 2 Data Write 2 Data Read 2 Data Write 2 Direct Branch 2 Direct Branch 2 Data Write 2 Data Write	Address: 0xl Address: 0xl Address: 0xl Address: 0xl Address: 0xl Address: 0xl	<pre>int mcount = 1007 volatile int numberArray[PERFNUM]; volatile int tempArray[FERFNUM]; index = 0;</pre>	0418 nove v #\$648,r6 nop doen3 #\$c8			
2 Data Read 2 Data Write 2 Direct Branch 2 Direct Branch 2 Data Read 2 Data Write	Address: 0x(Address: 0x(Address: 0x(Address: 0x(Address: 0x(Address: 0x(<pre>volatile int numberArray[PERFNUM]; volatile int tempArray[PERFNUM]; index = 0;</pre>	0418 nove.v #\$648.r6 nop doen3 #\$c8 0424 doen3 #\$c8			
2 Data Write 2 Direct Branch 2 Direct Branch 2 Data Read 2 Data Write	Address: 0x0 Address: 0x0 Address: 0x0 Address: 0x0 Address: 0x0	<pre>volatile int tempArray[FERFNUM]; index = 0;</pre>	doen3 #\$c8			
2 Direct Branch 2 Direct Branch 2 Data Read 2 Data Write	Address: 0x0 Address: 0x0 Address: 0x0	index = 0;	doen3 #\$c8			
2 Direct Branch 2 Data Read 2 Data Write	Address: 0x(Address: 0x(index = 0;	0424 depature2 (T109+0			
2 Data Read 2 Data Write	Address: 0x(Index - 0,				
2 Data Write	HUU 600. UK		0424 ddsecups (1100+0			
2 Long write	Address' Ovi	while(nCount)	0420 adda 10.3p			
2 Data Read	Address: Ovl	1	adda #=Sc sp r0			
2 Data Write	Address: Ovi		0422 adda \$-\$648 ep r			
2 Data Read	Address: Ovf	templirrav[index] = numberlrrav[nCour	adda \$_\$328 cp.r			
2 Data Write	Address: Ovi	support of (state) - support of other	043e nove 1 (ep-58) r			
2 Direct Branch	Address: 0x1	numberArray[ncounc] = numberArray[index	0440 move 1 (rfl) d2			
2 Direct Branch	Address: Oxf	numberArray[index] = tempArray[index];	0442 as12a r1			
2 Data Read	Address: Oxf	index++;	0444 adda m2 m1			
2 Data Minite	Address: Oxf	1	0446 more 1 d2 (r1)			
2 Data Read	Address: Ovf		0440 move 1 (sp-\$8) r			
2 Data Write	Address: Oxf		0440 8070.1 (3) 707.1			
2 Data Read	Address: Oxf		044c asl2a rl			
2 Data Write	Address: 0x(void PerfTest2 (void)	044e adda r3 r1			
2 Direct Branch	Address: 0xl	4	0450 nove 1 (r1) d3			
2 Direct Branch	Address: 0xf		0452 nove 1 d3 (r0)=			
2 Data Read	Address: 0vl	molatile int index:	0454 nove 1 (ep=88) r			
2 Data Moto	Address: Ovl -	Volacile inc index,	0454 move 1 (ep=68) r			
- Data Vinte	Pilder bass. over +	int ncount = 200;	0400 more.1 (ap vo),1			
	<u></u>	1				
213.161.85::jgriner						
006-09-14 15:32:04 Loading manifest for data file: D:\CW Analysis Tools-mi\analysis\1.0.0\workspace\test\datafiles\bwic 1.datafi						
2006-09-14 15:32:04 Data File (Dt/CN Analysis Tools-milanalysis). 0.0/worksnace/test/datafile//bujc 1 datafile). Start loading ma						
I ISISEISI BADA FILE (BI	ton Analysis is	ola-mi(umalysis(1.0.0(workspace(cest(ducalites(mwic_	inducative). Bod of loading i			
2006-09-14 15:32:09 Data File [D:\CW_analysis_Tools-mi\analysis\1.0.0\workspace\test\datafiles\hwic_1.datafile]: End of loading m						
15:32:07 Create data f	ile: D:\CW_Anal	ysis_Tools-ml\analysis\1.0.0\workspace\test\datafiles	<pre>>\hwic_0.datafile.</pre>			
15:32:08 Data File [D:	\CW Analysis To	ols-ml\analysis\1.0.0\workspace\test\datafiles\hwic C	.datafile]: Start loading (
			reasoning, reasoning			
	2 Data Yimo 2 Data Yimo 2 Data Yine 2 Dat	10 Martined 10 Martin	I Diverband			

Development Board •Includes MSC8156 device



Multicore StarCore® DSP Roadmap





MSC8156 Summary

- OEMs can take full advantage of LTE capabilities today
 - 6 GHz raw performance with fully programmable cores
 - Embedded Freescale MAPLE-B technology that accelerates Turbo and Viterbi decoding, Fast Fourier Transform (FFT), Inverse Fast Fourier Transform (IFFT), Discrete Fourier Transform (DFT), and Inverse Discrete Fourier Transform (IDFT) operations currently performed in FPGA or custom ASIC devices
 - Contains two configurable RISC engines, can be reprogrammed to accommodate updates
 - Supports legacy 3G technologies as well as the newest wireless standards the same DSP can be used for multiple technologies or in a multistandard basestations
- Manufactured in 45nm process technology
 - Significantly increases performance and design energy efficient solution while integrating more functionality
 - More cost effective solution, to design small form factor channel cards that take up less space, increased functionality and consumes less power giving the opportunity to provide differentiated and competitive solutions for their customers
- Reduces chip count and eliminates the need to attach costly, customized and power hungry dedicated devices
 - The combination of 6 cores DSP plus baseband accelerator into a single SoC can be used for final and cost optimized system production





Additional Information





Freescale ™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2008.



PSIF : Programmable System Interface TVPE : Turbo/Viterbi Processing Engine FFTPE : FFT Processing Engine DFTPE : DFT Processing Engine



Related Session Resources

Session Location – Online Literature Library

http://www.freescale.com/webapp/sps/site/homepage.jsp?nodeId=052577903644CB

Sessions

Session ID	Title

Demos

Pedestal ID	Demo Title		_





