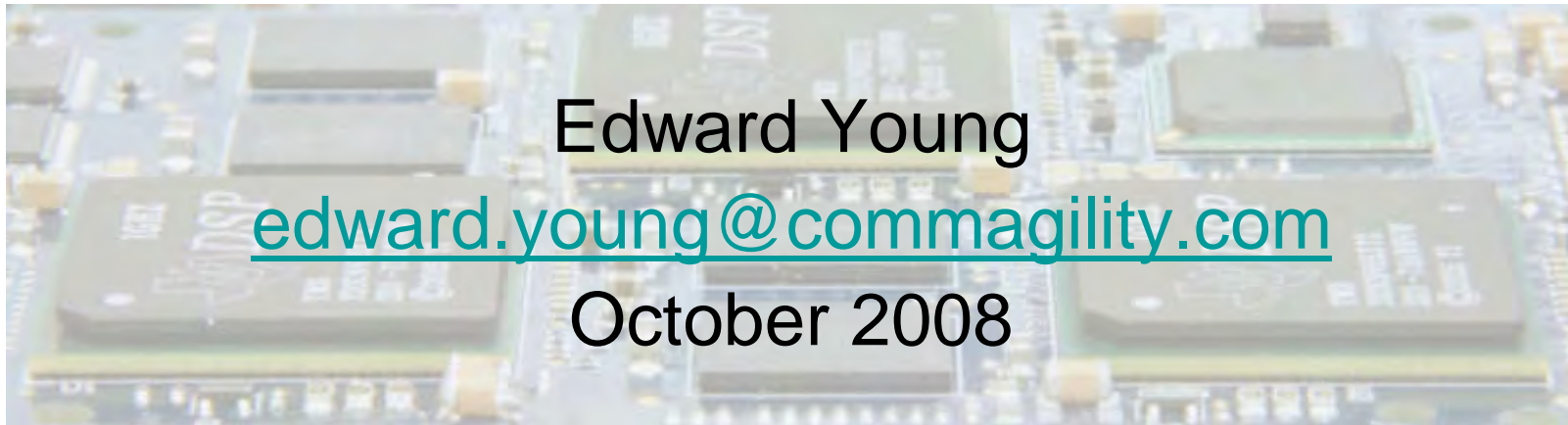


# Using Multicore DSPs in MicroTCA and AdvancedTCA Systems



Edward Young

[edward.young@commagility.com](mailto:edward.young@commagility.com)

October 2008

# About CommAgility



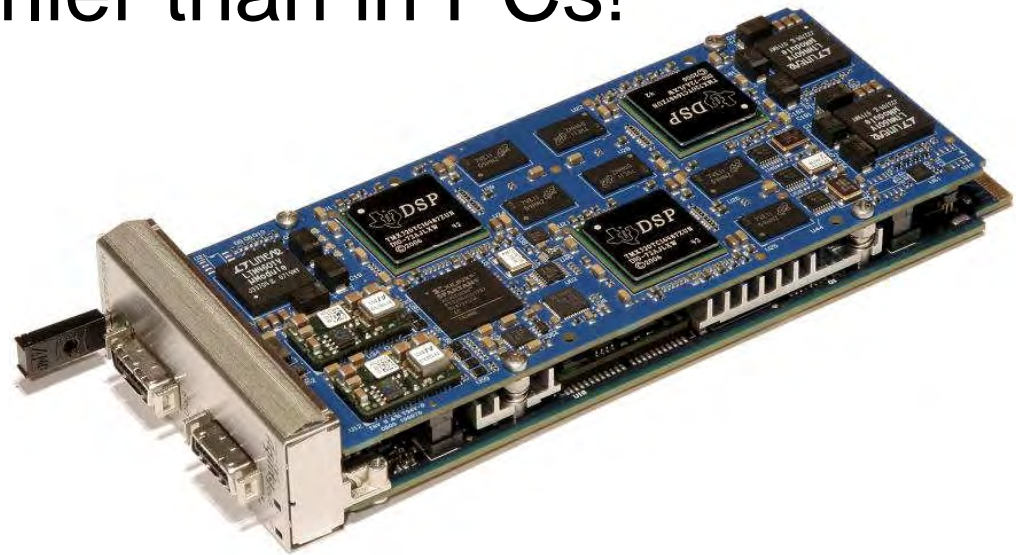
- CommAgility Ltd is a clear leader in signal processing AMC modules for wireless baseband applications, combining flexible antenna interfaces, the latest TI DSPs and Xilinx FPGAs, and high bandwidth on and off-card communications using Serial RapidIO and Ethernet. Customers around the world use CommAgility products to develop high performance applications in both wireless and non-wireless spaces, and recent designs include test equipment, trial systems and base stations for a wide range of wireless standards especially WiMax and LTE.
- Booth 521
- URL: [www.commagility.com](http://www.commagility.com)
- Email: [sales@commagility.com](mailto:sales@commagility.com)
- Tel: +44 1509 228866
- Innovation Centre, Epinal Way, Loughborough LE11 3EH, UK

# Contents

- Why Multicore DSP?
- Multicore Devices (TI)
- Benefits and Drawbacks
- ATCA and MicroTCA Hardware
- Example: Wireless Partitioning
- Conclusions

# Why MultiCore DSP?

- MultiProcessing is very common in DSP applications
- Power and space are important – this was an issue much earlier than in PCs!



# History

- DSP Multiprocessing has a long history
- 1990s
  - C40 and Sharc with built in link ports
  - RaceWay, SKYchannel etc at system level
  - Proprietary systems
- 2000s
  - Move to standards based
  - RapidIO designed for this

# Why Multiprocessing?



- DSP algorithms generally suit partition
  - More predictable than “control” code
  - Mostly block-based actions
  - Often multi-channel
  - Easy to parallelise - manually if needed
- FPGAs are pretty good at this too
  - A highly parallel resource
  - Increasing DSP features

# Power and Space

- Partitioning means less power per MIP
  - Power is not linear with clock speed
  - Assuming it can be done efficiently
- Multicore is better still
  - Reduces external pin use
- Multiprocessing is space hungry
  - High end systems – whole chassis!
  - Multicore helps this

# TI Devices Comparison

- TMS320C6455
  - Single core general purpose high end DSP
- TNETV3020
  - Multi-core voice processor
- TMS320TCI6487/88
  - Also now TMS320C6474
  - High end wireless processor
- All based on same C64x+ core

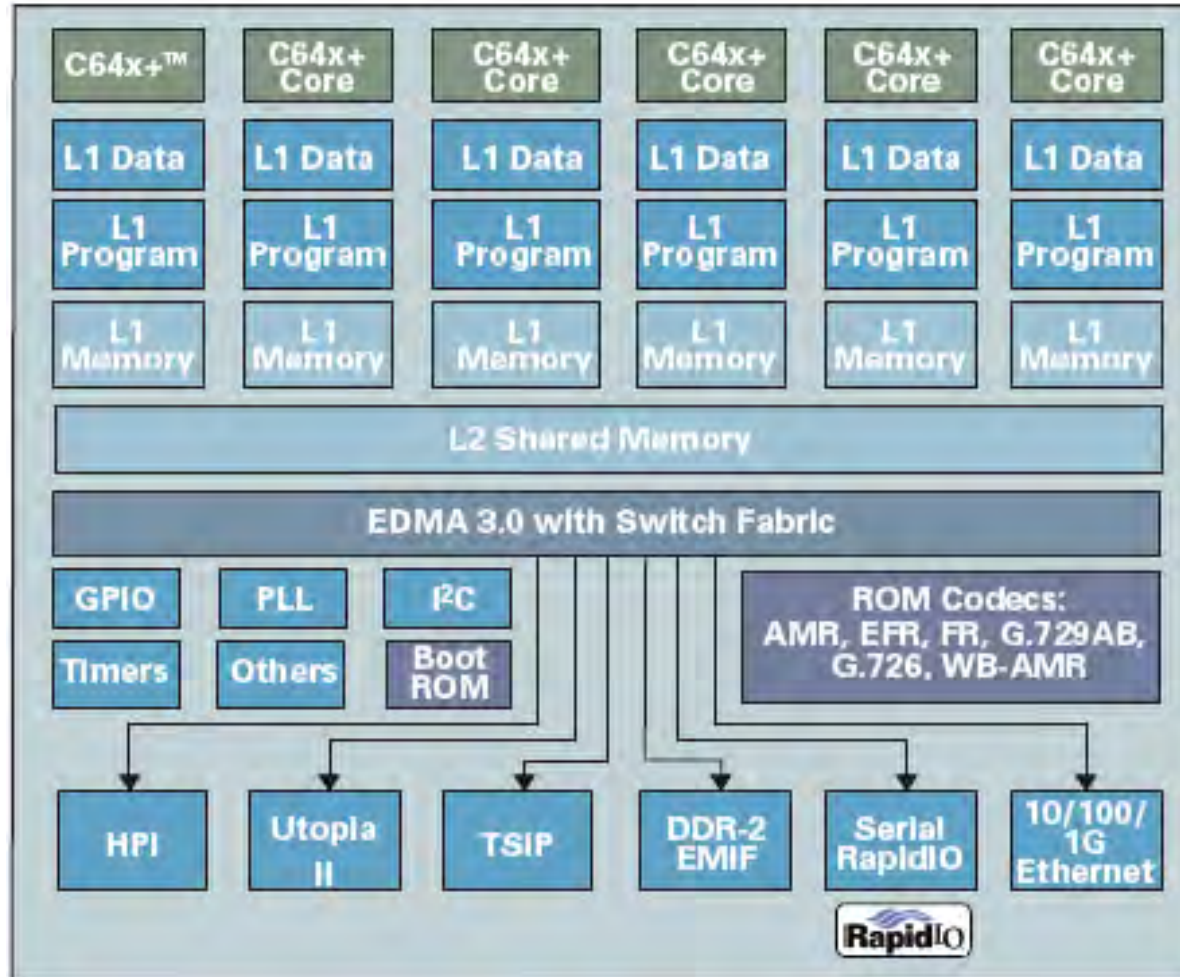
## C6455

- High end single core DSP
- Fixed point C64x+ core
- Up to 1.2 GHz
- Plenty of I/O
  - Parallel (EMIF), SRIO, Ethernet etc
  - DDR2 SDRAM

# TNETV3020

- Requirements
    - Aimed at multi-channel voice applications
    - Complex algorithms but repeated many times: easy to partition
    - Limited I/O BW – 1K x 64kbit/s = 64Mbps
  - Solution
    - Six cores at lower speed (500 MHz+) to reduce overall power
    - Relatively limited external bandwidth – focus on ethernet and telecom serial, some RapidIO
-

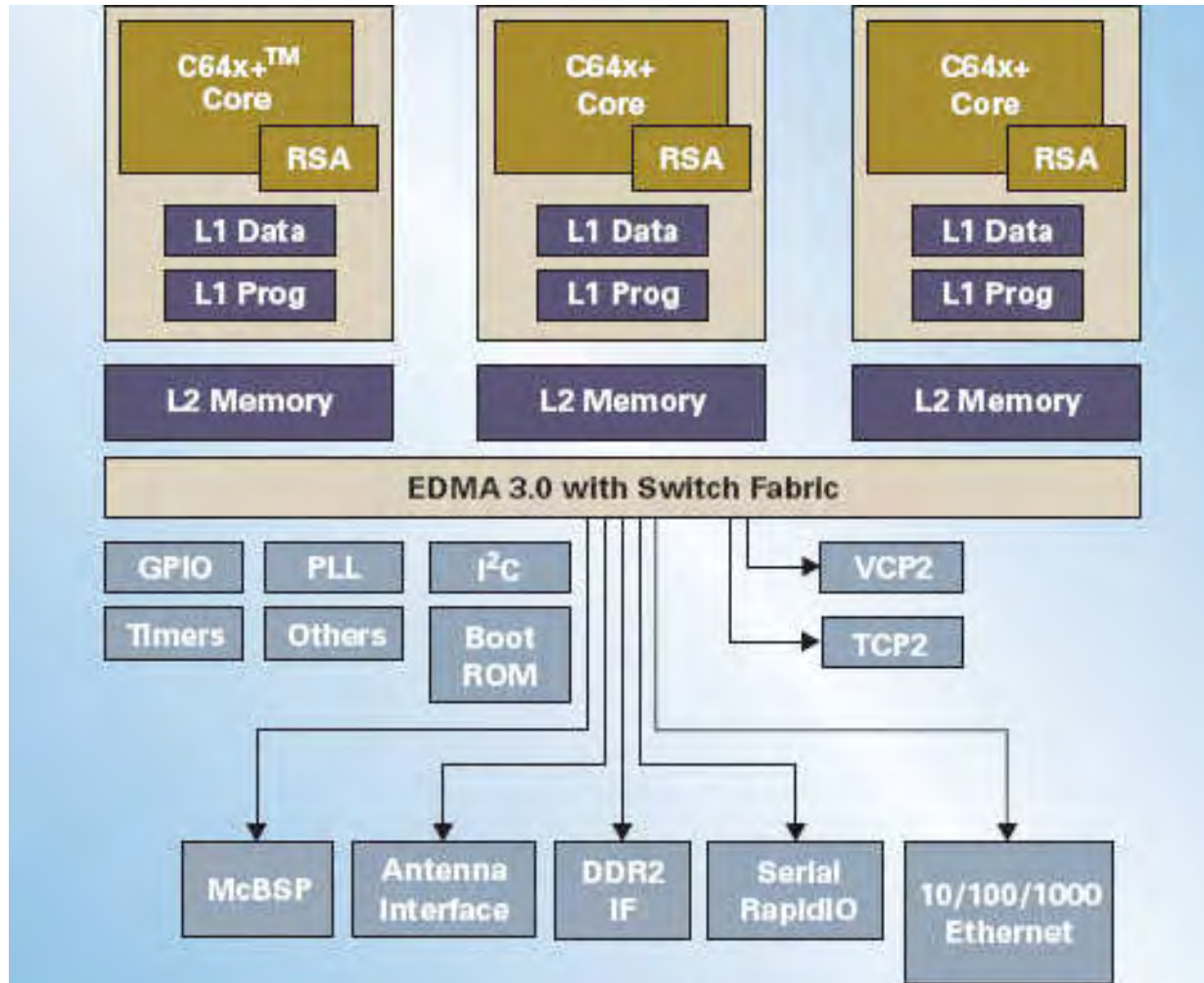
# TNETV3020 Diagram



## C6487/8 and C6474

- Requirements
  - Aimed at wireless infrastructure
  - Much harder algorithms to partition
  - Needs much higher I/O bandwidth
- Solution
  - Three cores at higher speed (1 GHz)
  - Larger shared memory/cache
  - More I/O BW: SRIO, Antenna i/fs, Ethernet
  - Specific accelerators – e.g. RSA, coding

# C6487/8 Diagram



# TI Semaphore Module



- Why have a Semaphore module?
  - **Management of shared resources**
  - **Software flags not effective, one core can modify while another is testing**
- What function does the Semaphore module provide?
  - **A method to control who accesses a shared resource**
  - **Atomic accesses for shared resources: Read-modify-write sequence**
- Features of the Semaphore module
  - **Contains 32 semaphores to be used within the system**
  - **Two methods of accessing a semaphore resource**
    - **Direct Access**
      - A core directly accesses a semaphore resource. If free, the semaphore will be granted. If not, the semaphore is not granted
      - Useful if the system can afford to poll for the semaphore
    - **Indirect access**
      - A core indirectly accesses a semaphore resource by writing to it. Once it is free an interrupt will notify the CPU that it is available.

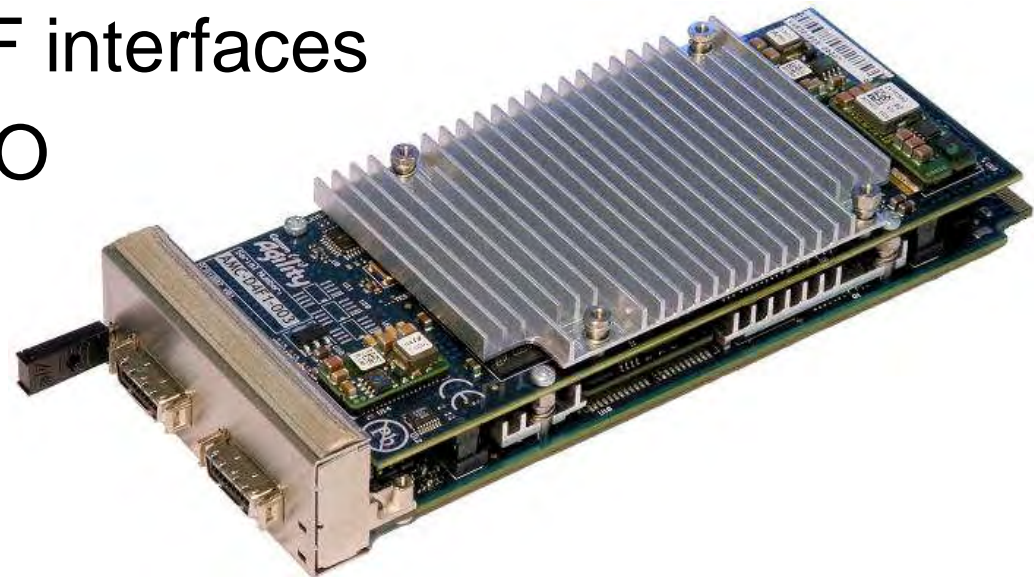
# Benefits and Drawbacks of DSP Multicore



- Benefits
  - Power, space and cost
  - Works well with DSP applications
- Drawbacks
  - Shared chip resources
  - Less I/O bandwidth per core
  - May be an issue in high end applications

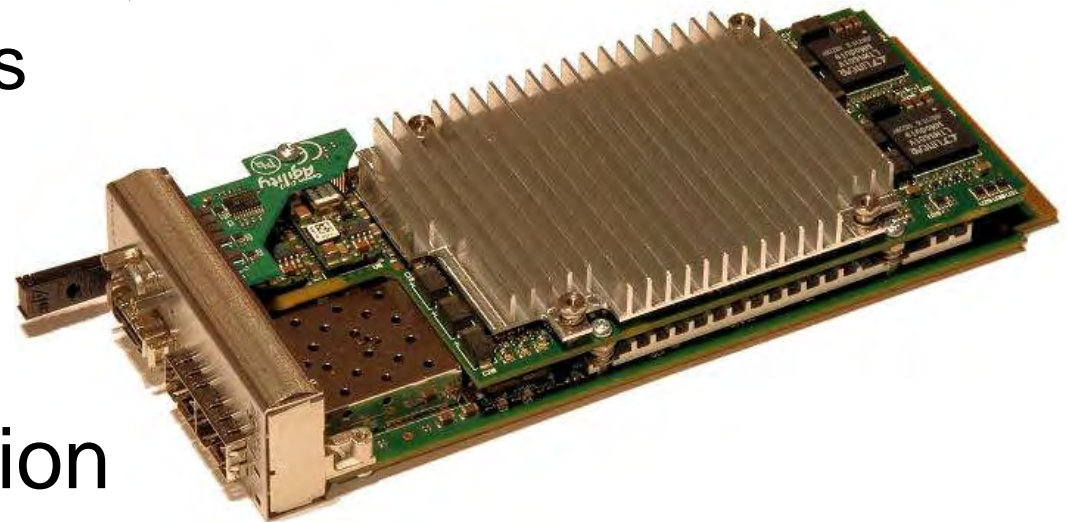
# Full Size AMC Example 1

- AMC-D4F1-1200: highest I/O
  - 4 x 1.2 GHz C6455 DSPs: 4800 MIPS
  - Virtex 4 FX100 FPGA
  - DSP-FPGA EMIF interfaces
  - Full 10Gbps SRIO
  - Ethernet, ....
- Typical use:  
wireless test



# Full Size AMC Example 2 *Comm (( )) Agility*

- AMC-3C87F: highest performance
  - 3 x 1 GHz C6487 DSPs: 9000 MIPS!
  - Virtex 5 up to SX95T/LX155T
  - 5Gbps SRIO to DSPs, no EMIF
  - Antenna interfaces
  - Space for 2 SFPs
  - Ethernet, ...
- Typical use:  
wireless base station



# ATCA Example

- RadiSys product using TNETV3020
- Aimed at high density voice
- Up to 20 x TNETV3020  
= 120 cores and 60000 MIPS!

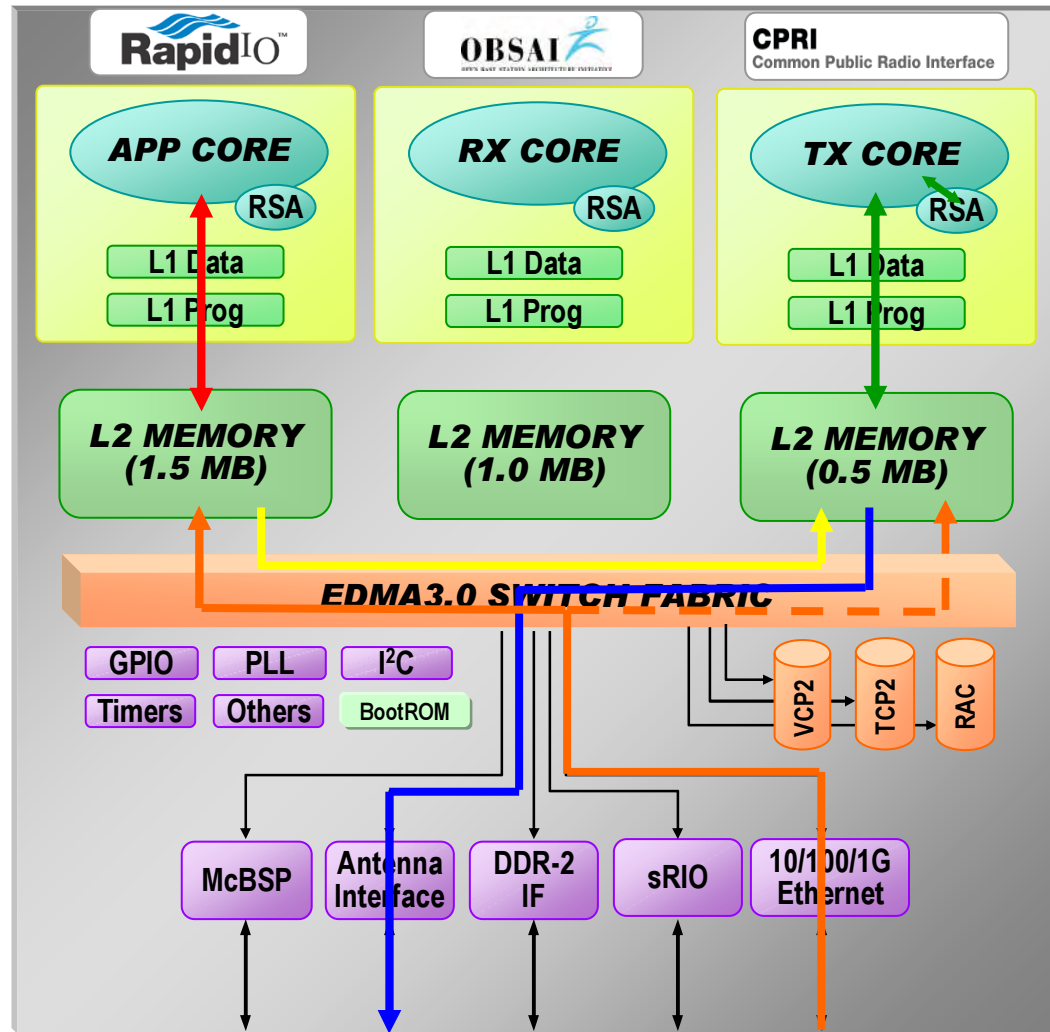


# Wireless Applications



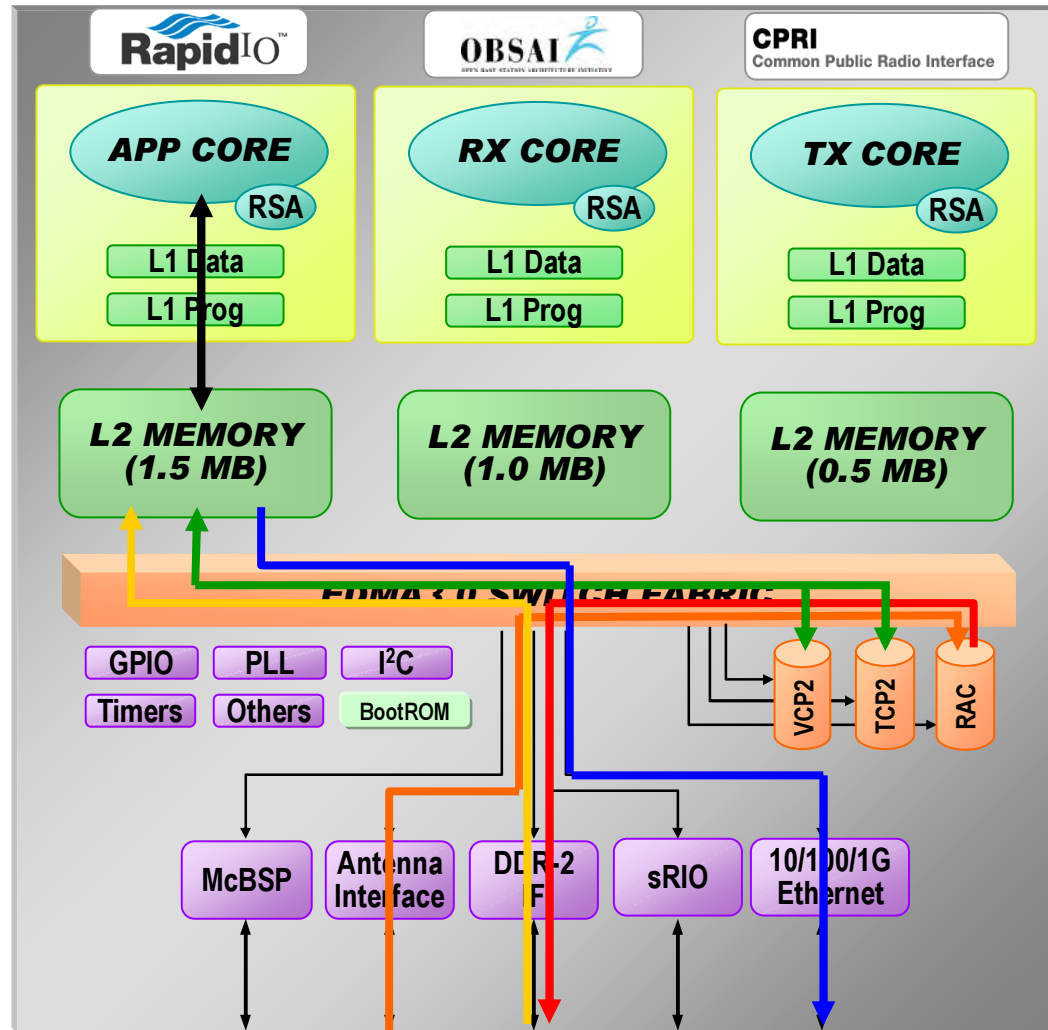
- 3G Wireless Example
  - Full implementation on a single DSP
  - Algorithm partitioning
- WiMAX Example
  - Multiple, Multicore DSPs with FPGA

# Transmit Data Flow



- Network data is received from the network (lub user plane) via the Network Interface and written to the Application Core Memory
- If TPC bits are generated on another device, they are written by the Network Interface to the Transmit Core memory.
- TX Symbol rate processing occurs using the App Core.
- Once complete, the App core will send the DL symbol data to the TX Core
- TX Chip-rate processing occurs in the TX Core
- The data is then sent to the Antenna Interface to be transmitted

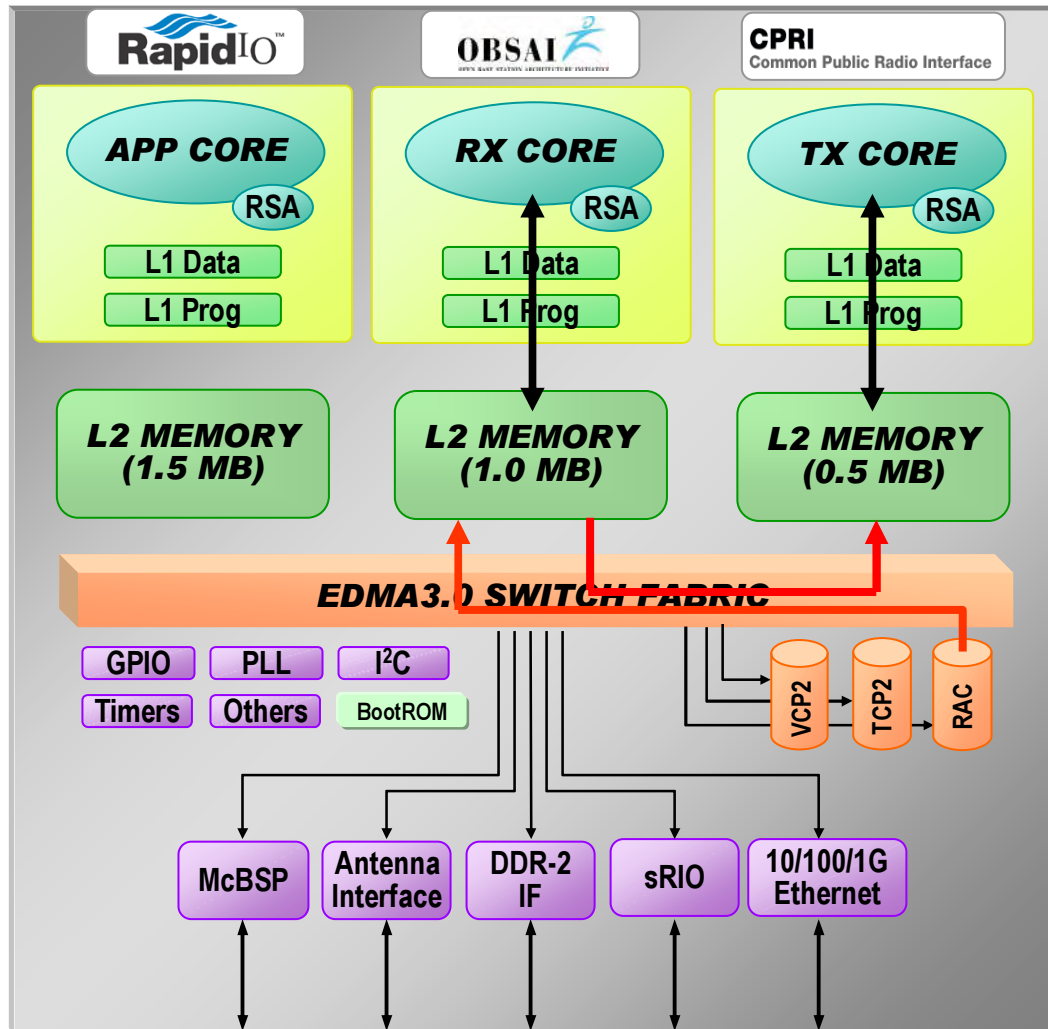
# Receive Data Flow



- Antenna data is received and transferred to RAC.
- On-Time Pre-MRC finger symbols are then copied to external memory attached to the DDR2 EMIF
- The Application Core reads the symbols from external memory and perform the symbol rate processing
- During the symbol-rate processing data is sent to the TCP/VCP for processing and decoded data is returned.
- User data is then sent to the network via the Network Interface

# Closed Loop Control

Comm (( ))  
**Agility**



- DPCCH de-spread symbols from received antenna data is transferred from RAC to the Receive Core
- TPC for uplink and decision for downlink TPC are transferred to the Transmit Core

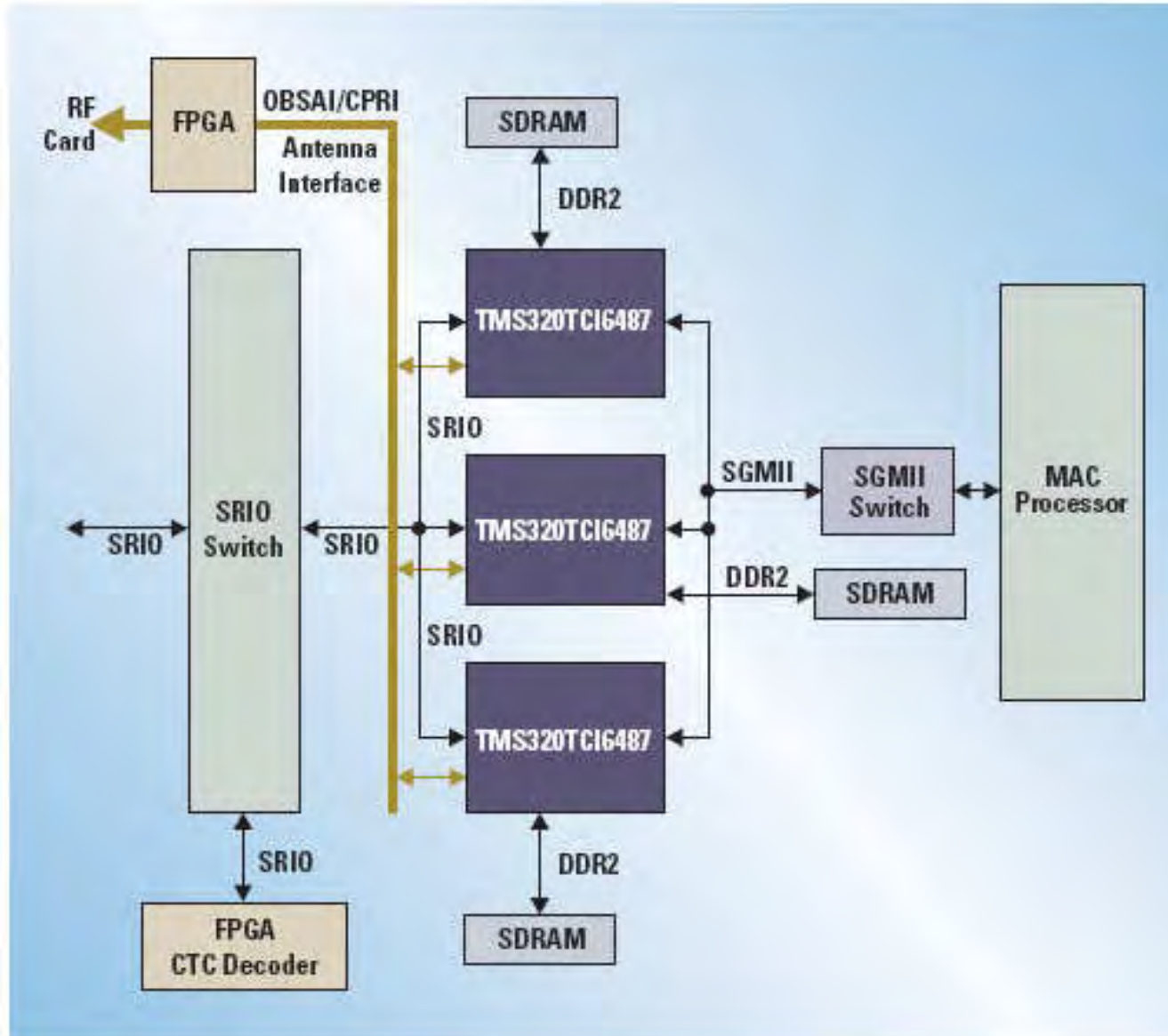
# Comments

- Manual algorithm partitioning for maximum efficiency
- Each core is doing different tasks
- Separate L2 cache for each core
- Different L2 cache sizes
- Use of co-processors

# WiMAX



- A complete 10-MHz, 2-antenna, 3-sector solution can be implemented with only three TCI6487 DSPs where each sector can support a full 10-MHz band with 2x2 MIMO
- An FPGA is used perform a CTC decoder function.
- The TCI6487 improves the cost and power per channel by accelerating key components of baseband processing like matrix math processing.



# Conclusions

- DSP Multicore is mainstream for the high end
  - Wireless, high density voice, video etc
  - A natural development from multi-processing
- Algorithm partitioning is still manual
  - Provides greatest optimisation
- Deployed solutions exist for MicroTCA & ATCA
  - Generally multi-processor
  - Single and multi-core are both deployed depending on application needs
  - May also have an FPGA for specific I/O and acceleration functions



**Thank you for listening!**

edward.young@commagility.com