Reevaluating the Latency Claims of 3D Stacked Memories

Daniel Chang[†], Gyung-su Byun[‡], Hoyoung Kim[§], Minwook Ahn[§], Soojung Ryu[§], Nam Kim[†], Michael Schulte[†]

[†] University of Wisconsin - Madison
[‡] University of West Virginia
[§] Samsung Electronics Company

dwchang@wisc.edu

Outline

- Background
  - What is 3D Technology?
  - Why DSPs?
- A Model for 3D DRAM
  - DRAM memory timings
  - Original 3D DRAM latency claim vs. our 3D DRAM model
- Experimental Framework
  - Simulation Infrastructure and Benchmarks
- Reevaluating 3D DRAM
  - Latency
  - Bandwidth
  - Power
- Conclusions and Future Work

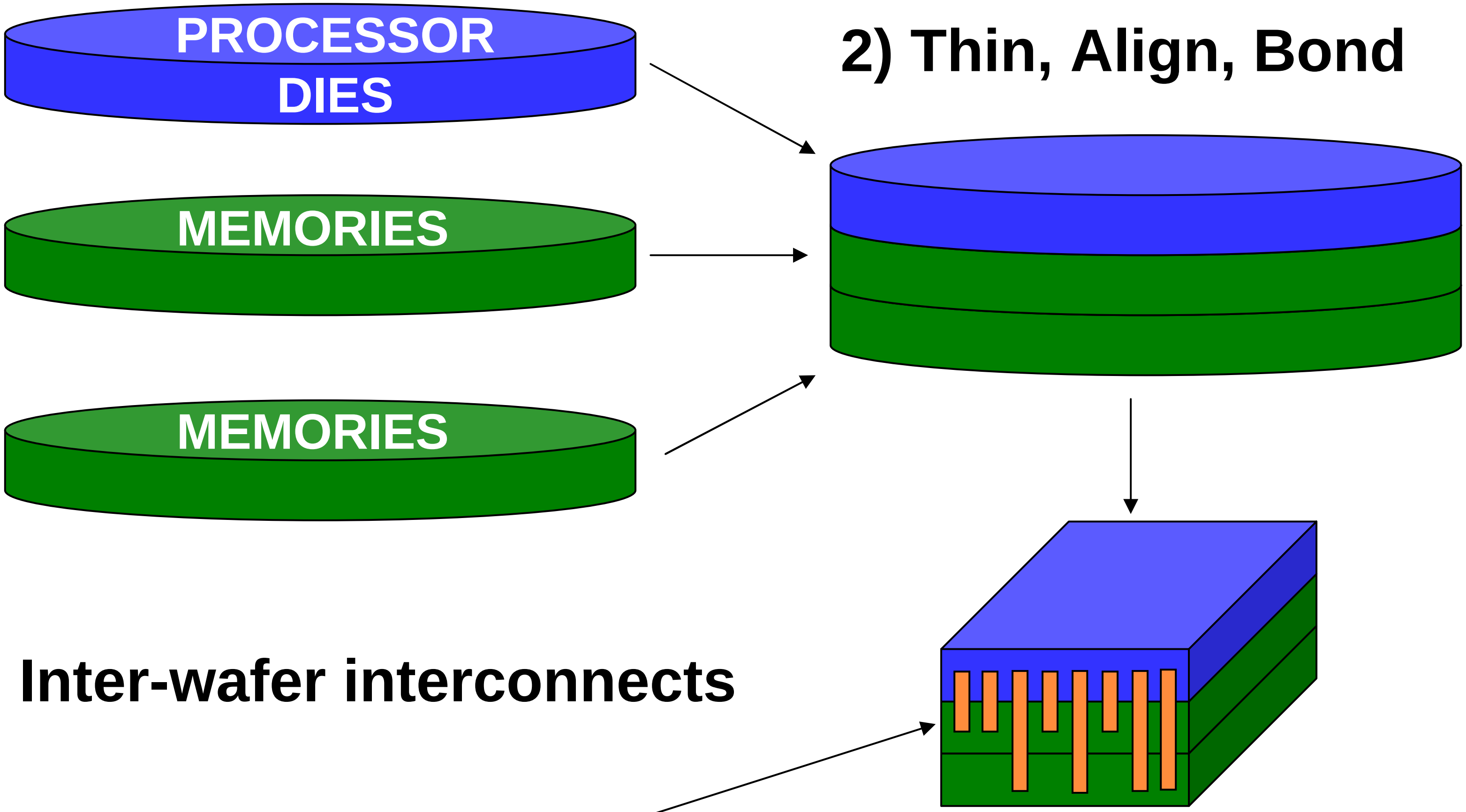
What is 3D Integration Technology?

- Multiple integrated chips vertically stacked
  - Increased device density
  - Greater routing flexibility and reduced wire lengths
- Three different techniques:

1. 3D packaging technology
   - Die-to-die technology
   - Low inter-die interconnect density
2. Transistor build-up 3D technology
   - Forms transistors inside on-chip interconnect layers
   - Not compatible with existing fabrication processes
   - Subject to severe process temperature constraints
3. Wafer-level, back-end-of-the-line compatible 3D technology
   - Wafer-to-wafer technology that uses through-silicon vias
   - Greatest interconnect density, but high cost
   - Allows integration of different technologies

Wafer Level 3D Technology

1) Fabricate separate wafers



2) Thin, Align, Bond

3) Inter-wafer interconnects

Through-Silicon Vias

3D Stacked Processor

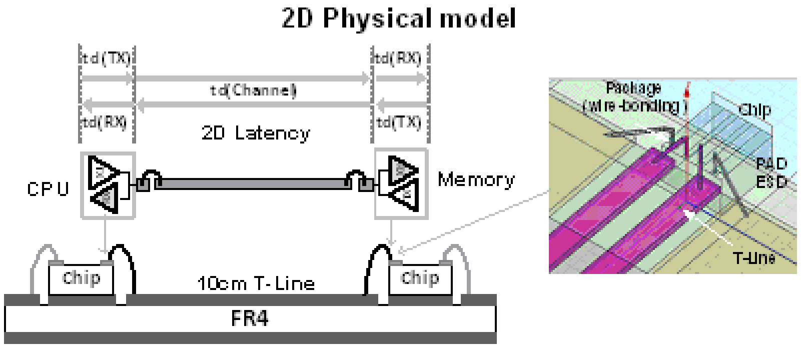
Motivation

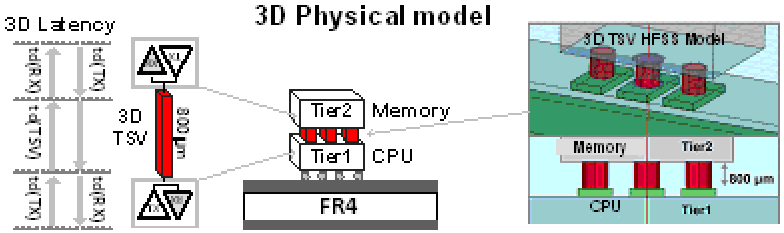
- 3D DRAM may directly address the memory wall problem
  - One of the main advantages is reduced access latency
  - However if this claim is inaccurate it could lead to sub-optimal memory architectures
- Why DSPs?
  - DSPs are playing an increasingly important role in modern computing systems
  - DSP applications are becoming more compute and memory intensive while requiring low energy consumption
    - Multimedia, computer vision, speech recognition, etc.
  - DSPs have small on-chip caches and there is a growing need for lower off-chip memory latency

DRAM Timing Model

- DRAM performance specified by various parameters
  - tRP - cycles needed to terminate access to a row and open another row
  - tRAS - cycles to access a certain row after sending row address
  - tRCD – cycles between opening a row and accessing col.
  - tCL - cycles to access a col. of data after sending col. address
- DRAM latencies vary depending on:
  - Data is in an already open row (tCL)
  - Data resides in a different row (tRP + tRCD + tCL)
- Additionally must include off-chip interconnect latency

A Model for 3D DRAM





- Widely accepted claim: 45 – 60% reduction in latency
  - Latency savings comes from going to TSV interconnects
  - DRAM bank organization remains the same
- 2D and 3D interconnects both still have:
  - Transmitter
  - Receiver
  - Transmission medium or channel

A Model for 3D DRAM

- Modeled 2D FR-4 and 3D TSV channels with Spectre circuit simulator
  - Latency values based on circuit and physical models including a 2D package (wire-bonding), ESD, and I/O pad parasitic components

Component	2D T-Line (10 cm)	2D T-Line (20 cm)	3D TSV
td (TX)	69 ps	70 ps	69 ps
td (channel)	620 ps	1,210 ps	21 ps
td (RX)	235 ps	242 ps	220 ps
Total	**924 ps**	**1,522 ps**	**310 ps**

- TSV latency savings ~600 – 1,200 ps
  - For a random DRAM access this is a < 2% latency reduction

Simulation Infrastructure

- Trimaran simulator running within EPIC explorer
  - Cycle accurate, parameterizable compilation and performance evaluation tool for embedded and VLIW architectures
- M5elements
  - Cache simulator that allows the Simu simulator to use the memory subsystem of the M5 simulator
  - Used to gather detailed memory statistics
- CACTI
  - Cache and DRAM simulator
  - Used to obtain memory latencies and power and area estimates
- DRAMSim2
  - Trace-based DRAM simulator
  - Used to obtain memory latencies specific to each application

DRAM Model

- Modeled 512 MB DDR2 DRAM using 65 nm in CACTI
  - 56 ns latency very close to Micron DDR2 spec sheet [1]
    - This is for a random DRAM accesses
- 2D DRAM
  - Ran application-specific memory traces in DRAMSim2 with timing parameters generated by CACTI
    - 38 and 40 ns (Our application's accesses are not always random)
- Original 3D DRAM
  - Took the 2D DRAM latencies and reduced them by 45%
    - 21 and 22 ns
- Our 3D DRAM
  - Took the 2D DRAM latencies and reduced them by 1 ns
    - 37 and 39 ns
- Used these latencies for each Trimaran simulation

[1] Micron. 512 MB: x4, x8, x16 DDR2 SDRAM Features, 2004. [Online] Available: http://download.micron.com/pdf/datasheets/dram/ddr2/512MbDDR2.pdf

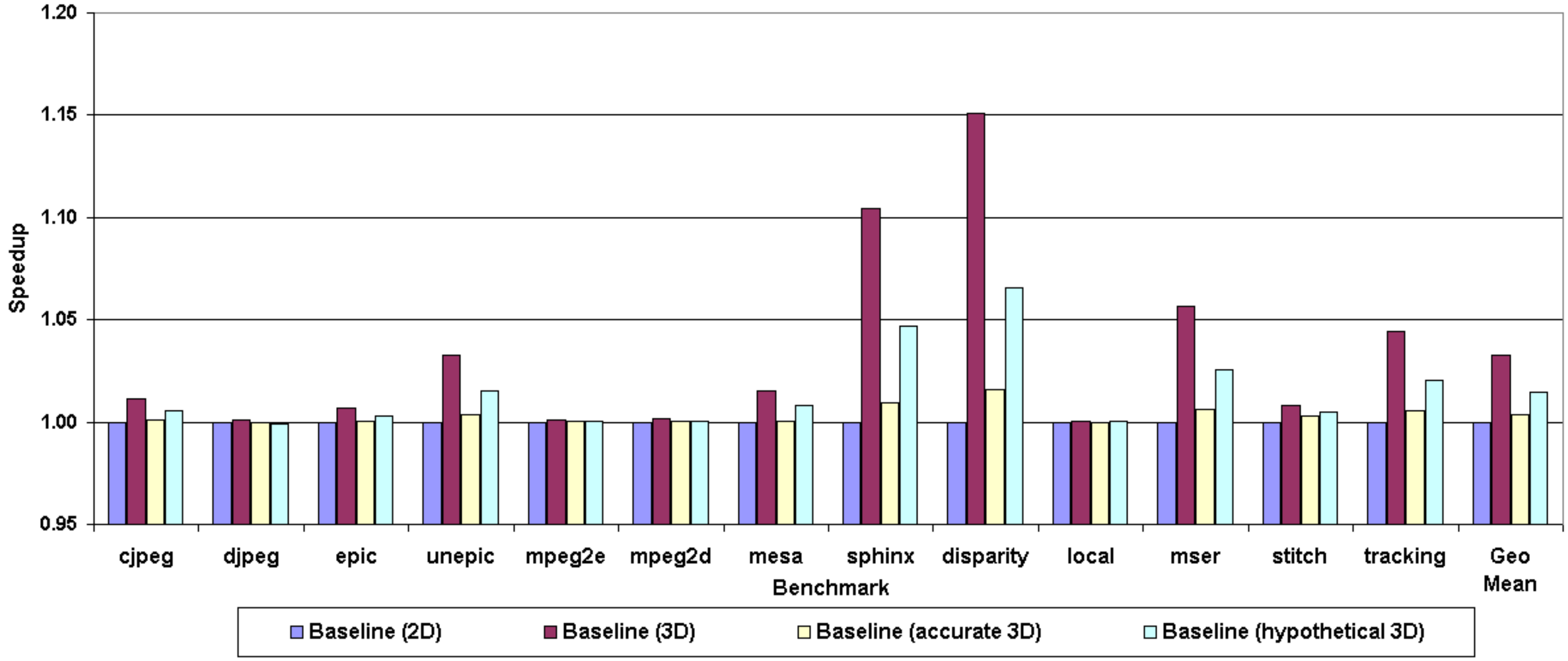
Benchmarks

- DSP applications commonly run on mobile devices
  - Encompass a variety of application types from multimedia processing, 3D graphics and speech recognition
- Seven MediaBench benchmarks:
  - cjpeg, djpeg, epic, unepic, mpeg2enc, mpeg2dec and mesaosdemo (3D graphics)
- One SPEC CPU2006 benchmark:
  - 482.sphinx3 (speech recognition)
- Five San Diego Vision benchmarks:
  - disparity, localization, mser (face recognition), stitch and tracking

Experimental Methodology

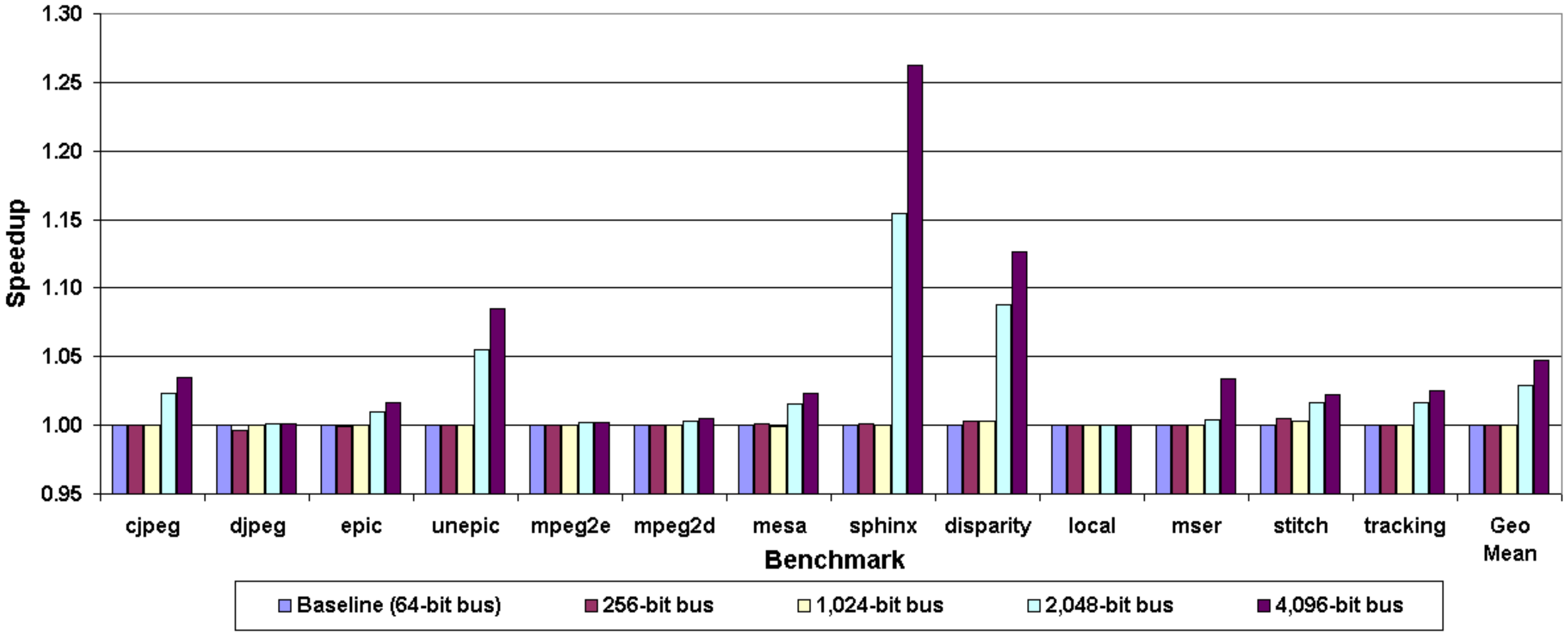
- Goal: To reevaluate the impact of 3D DRAMs on DSPs
- Baseline similar to dual-core TI C67x (8-way VLIW)
  - 1 GHz, 32 KB L1D, 32 KB L1I, 256 KB L2, six integer units and two floating point units
- Quantify the performance of:
  - Baseline DSP with 2D DRAM (38 and 40 ns)
  - Baseline DSP with original 3D DRAM (21 and 22 ns)
    - Allows us to quantify the potential performance benefits of the original 3D DRAM latency claim
  - Baseline DSP with our 3D DRAM (37 and 39 ns)
  - Baseline DSP with hypothetical 3D DRAM (30 ns)
  - Baseline DSP with increased main memory bandwidth
    - Increased bandwidth from 64 to 256 and 1,024 bits
    - Increased bandwidth and L2 line size to 2,048 and 4,092 bits

Results – Reduced Access Latency



- Original 3D DRAM model achieves a 3.2% average speedup
  - Unepic, sphinx, disparity and mser exhibit largest speedup (3.3% - 15.1%)
- Accurate 3D DRAM model achieves a < 1.0% average speedup
- Hypothetical 3D DRAM model achieves a 1.5% average speedup
  - Speedups may be possible with new optimizations

Results – Increased Bandwidth



- Increasing bandwidth to 256 and 1,024 bits shows little improvement
- Increasing bandwidth and L2 line size to 2,048 bits achieves a 2.9% average speedup
- Increasing bandwidth and L2 line size to 4,096 bits achieves a 4.7% average speedup
  - unepic, sphinx and disparity exhibit largest speedup (8.5% – 26.3%)

Results – Power Consumption

- TSVs consume 11.2x less I/O power per bit transfer than off-chip metal interconnects
  - 8.71 mW per bit in 2D vs. 0.78 mW per bit in 3D (both 800 MHz)
- However, the power reduction of one component does not always result in a decrease in power
  - When memory width is increased beyond 11x, power increases
- We increased the main memory bus by up to 64x
  - 64 bits up to 4,096 bits
  - A 2D 64-bit bus would consume 557 mW
  - A 3D 4,096-bit bus would consume 3.2 W
    - A 5.7x increase in power consumed by the main memory bus
- However, we are sending more data
  - May allow designers to lower the transfer frequency to save power
  - Or simply keep the 64-bit bus and go to 3D TSVs to save power

Conclusions

- Widely accepted 3D DRAMs save 45 – 60% access latency by going to TSVs
- Through circuit-level simulations we find the savings to be much smaller (~600 – 1,200 ps)
  - Overall memory bank architecture remains the same
  - 3D interconnects still require transmitter, receiver and transmission medium
- 3D TSVs can increase main memory bandwidth and system performance
  - Between 8.5% – 26.3% speedup on some applications
  - However, power consumption may become a problem with very large memory bus widths

Future Work

- Future work aims to:
  - Look for better ways to take advantage of 3D technology and reduce latency
    - Can we remove some components to reduce latency?
    - Can the memory organization be redone to take better advantage of high bandwidth?
  - Dynamically main memory bandwidth scheduling algorithms
    - Increase main memory bandwidth to increase performance
    - Decrease main memory bandwidth to lower power

Questions?