

Heterogeneous Thinking

Michael O'Boyle
University of Edinburgh

icsa

Institute for Computing
Systems Architecture



Rethinking the Hardware/Software Contract

Michael O'Boyle
University of Edinburgh

icsa

Institute for Computing
Systems Architecture



Rethinking the hardware/software contract

Heterogeneity

Great

- Dark silicon etc
- Hardware avoids abstraction tax

No free lunch

- Software has to pick up the tab

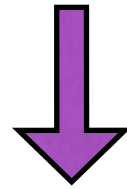
Hardware/software contract

- defined by an API: the ISA

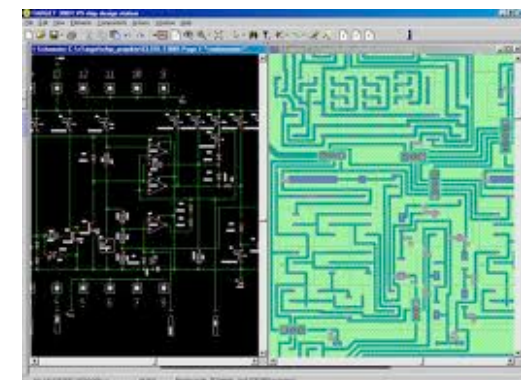
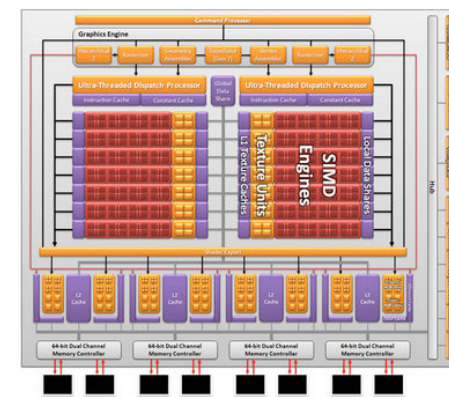
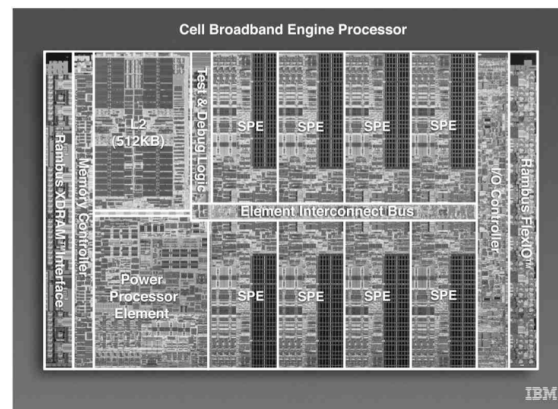
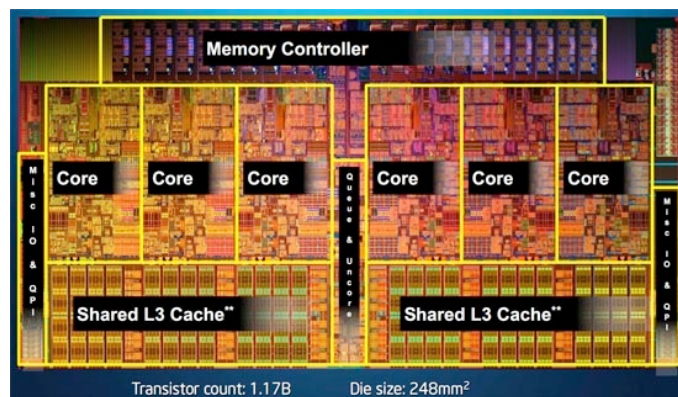
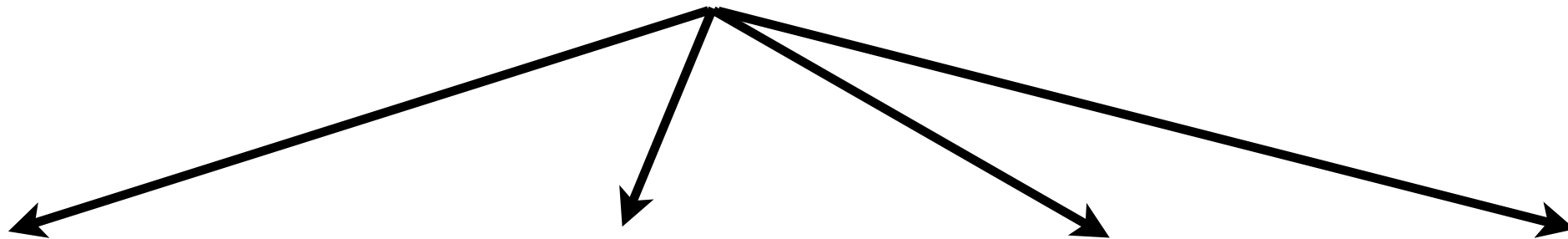
ISA no longer the API for heterogeneity

Heterogeneity: Hardware Zoo

Application



Parallel Language



GPU-Accelerated Libraries

GPU-Accelerated libraries provide highly-optimized algorithms and functions you can incorporate into your applications, with minimal changes to your existing code. Many support drop-in compatibility to replace industry standard CPU-only libraries such as MKL, IPP, FFTW and widely-used libraries. Some also feature automatic multi-GPU performance scaling.



AmgX

A simple path to accelerated core solvers, providing up to 10x acceleration in the computationally intense linear solver portion of simulations, and is very well suited for implicit unstructured methods.



cuDNN

NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks. It is designed to be integrated into higher-level machine learning frameworks.



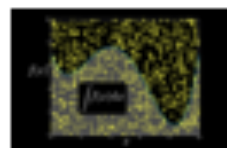
cuFFT

NVIDIA CUDA Fast Fourier Transform Library (cuFFT) provides a simple interface for computing FFTs up to 10x faster, without having to develop your own custom GPU FFT implementation.



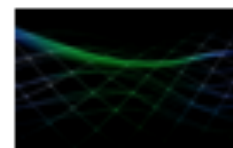
Index Framework

NVIDIA Index Framework is a real-time scalable visualization plug-in for ParaView.



cuRAND

The CUDA Random Number Generation library performs high quality GPU-accelerated random number generation (RNG) over 6x faster than typical CPU only code.



CUDA Math Library

An industry proven, highly accurate collection of standard mathematical functions, providing high performance on NVIDIA GPUs.



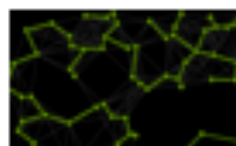
Thrust

A powerful, open source library of parallel algorithms and data structures. Perform GPU-accelerated sort, scan, transform, and reductions with just a few lines of code.



NVBIO

A GPU-accelerated C++ framework for High-Throughput Sequence Analysis for both short and long read alignment.



nvGRAPH

nvGRAPH Analytics Library is a GPU-accelerated graph analytics library.



GIE

NVIDIA GPU Inference Engine is a high performance neural network inference library for deep learning applications



NPP

NVIDIA Performance Primitives is a GPU accelerated library with a very large collection of 1000's of image processing primitives and signal processing primitives.



FFmpeg

FFmpeg is a popular open-source multi-media framework with a library of plugins that can be applied to various parts of the audio and video processing pipelines.



NVIDIA VIDEO CODEC SDK

Accelerate video compression with the NVIDIA Video Codec SDK. This SDK includes documentation and code samples that illustrate how to use NVIDIA's NVENC and NVDEC hardware in GPUs to accelerate encode, decode, and transcode of H.264 and HEVC video formats.



HiPLAR

HiPLAR (High Performance Linear Algebra in R) delivers high performance linear algebra (LA) routines for the R platform for statistical computing using the latest software libraries for heterogeneous architectures.



OpenCV

OpenCV is the leading open source library for computer vision, image processing and machine learning, and now features GPU acceleration for real-time operation.



Geometry Performance Primitives(GPP)

GPP is a computational geometry engine that is optimized for GPU acceleration, and can be used in advanced Graphical Information Systems (GIS), Electronic Design Automation (EDA), computer vision, and motion planning solutions.



CHOLMOD

GPU-accelerated CHOLMOD is part of the SuiteSparse linear algebra package by Prof. Tim Davis. SuiteSparse is used extensively throughout industry and academia.



CULA Tools

GPU-accelerated linear algebra library by EM Photonics, that utilizes CUDA to dramatically improve the computation speed of sophisticated mathematics.



MAGMA

A collection of next gen linear algebra routines. Designed for heterogeneous GPU-based architectures. Supports current LAPACK and BLAS standards.



IMSL Fortran Numerical Library

Developed by RogueWave, a comprehensive set of mathematical and statistical functions that offloads work to GPUs.



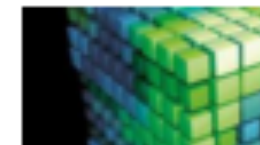
Parallelization

Library for sparse iterative methods with special focus on multi-core and accelerator technology such as GPUs.



Triton Ocean SDK

Triton provides real-time visual simulation of the ocean and bodies of water for games, simulation, and training applications.



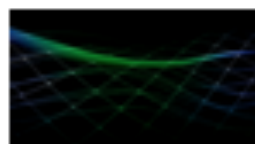
cuBLAS

NVIDIA CUDA BLAS Library (cuBLAS) is a GPU-accelerated version of the complete standard BLAS library that delivers 6x to 17x faster performance than the latest MKL BLAS.



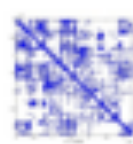
ArrayFire

Comprehensive, open source GPU function library. Includes functions for math, signal and image processing, statistics, and many more. Interfaces for C, C++, Java, R and Fortran.



cuSOLVER

A collection of dense and sparse direct solvers which deliver significant acceleration for Computer Vision, CFD, Computational Chemistry, and Linear Optimization applications




cuSPARSE

NVIDIA CUDA Sparse (cuSPARSE) Matrix library provides a collection of basic linear algebra subroutines used for sparse matrices that delivers over 8x performance boost.

Good performance is hard to get even with well defined parallel language CUDA/OpenCL









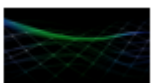

GPU-Accelerated Libraries

GPU-Accelerated libraries provide highly-optimized algorithms and functions you can incorporate into your applications, with minimal changes to your existing code. Many support drop-in compatibility to replace industry standard CPU-only libraries such as MKL, IPP, FFTW and widely-used libraries. Some also feature automatic multi-GPU performance scaling.

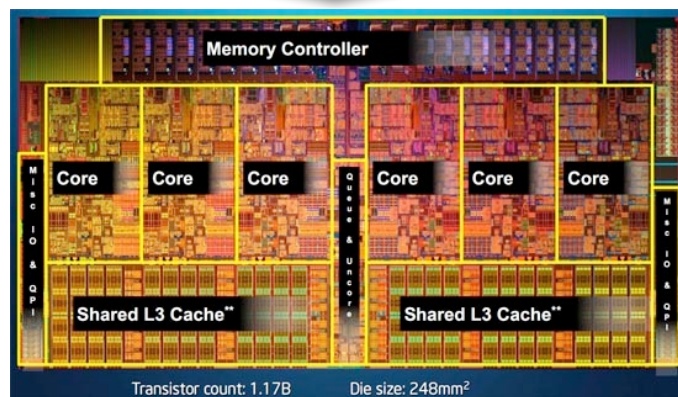
 <p>AmgX A simple path to accelerated core solvers, providing up to 10x acceleration in the computationally intense linear solver portion of simulations, and transparently scaled for multi-GPU.</p>	 <p>cuDNN NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks. It is designed to be integrated into higher-level machine learning frameworks.</p>	 <p>cuFFT NVIDIA CUDA Fast Fourier Transform Library (cuFFT) provides a simple interface for computing FFTs up to 10x faster, without having to develop your own custom GPU code.</p>	 <p>Index Framework NVIDIA Index Framework is a real-time scalable visualization plug-in for ParaView.</p>	 <p>cuRAND The CUDA Random Number Generation library performs high quality GPU-accelerated random number generation (RNG) over 8x faster than typical CPU-only code.</p>	 <p>CUDA Math Library An industry proven, highly accurate collection of standard mathematical functions, providing high performance on NVIDIA GPUs.</p>	 <p>Thrust A powerful, open source library of parallel algorithms and data structures. Perform GPU-accelerated sort, scan, transform, and reductions with built-in functions and functors.</p>	 <p>NVBIO A GPU-accelerated C++ framework for High-Throughput Sequence Analysis for both short and long read alignment.</p>
---	--	--	--	--	---	--	---

Rather than building a new optimising compiler for each platform

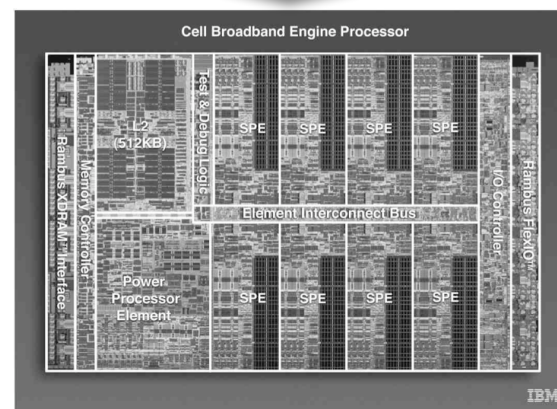
Pick the best Library/API/DSL and fit the code to it

 <p>CHOLMOD GPU-accelerated CHOLMOD is part of the SuiteSparse linear algebra package by Prof. Tim Davis. SuiteSparse is used extensively throughout industry and academia.</p>	 <p>CULA Tools GPU-accelerated linear algebra library by EM Photonics, that utilizes CUDA to dramatically improve the computation speed of sophisticated mathematics.</p>	 <p>MAGMA A collection of next-gen linear algebra routines. Designed for heterogeneous GPU-based architectures. Supports current LAPACK and BLAS standards.</p>	 <p>IMSL Fortran Numerical Library Developed by RogueWave, a comprehensive set of mathematical and statistical functions that offloads work to GPUs.</p>	 <p>oralution Library for sparse iterative methods with special focus on multi-core and accelerator technology such as GPUs.</p>	 <p>Triton Ocean SDK Triton provides real-time visual simulation of the ocean and bodies of water for games, simulation, and training applications.</p>	 <p>cuBLAS NVIDIA CUDA BLAS Library (cuBLAS) is a GPU-accelerated version of the complete standard BLAS library that delivers 8x to 17x faster performance than the latest MKL BLAS.</p>	 <p>ArrayFire Comprehensive, open source GPU function library. Includes functions for math, signal and image processing, statistics, and many more. Interfaces for C, C++, Java, R and Fortran.</p>
 <p>cuSOLVER A collection of dense and sparse direct solvers which deliver significant acceleration for Computer Vision, CFD, Computational Chemistry, and Linear Optimization applications.</p>	 <p>cuSPARSE NVIDIA CUDA Sparse (cuSPARSE) Matrix library provides a collection of basic linear algebra subroutines used for sparse matrices that delivers over 8x performance boost.</p>						

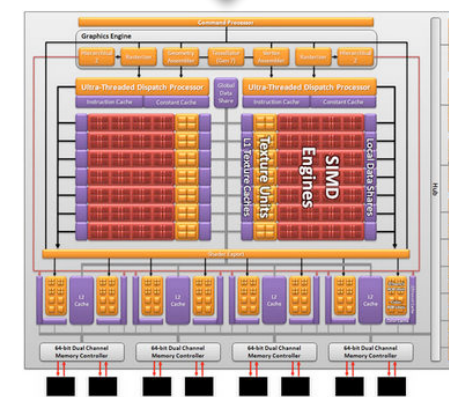
Legacy Program



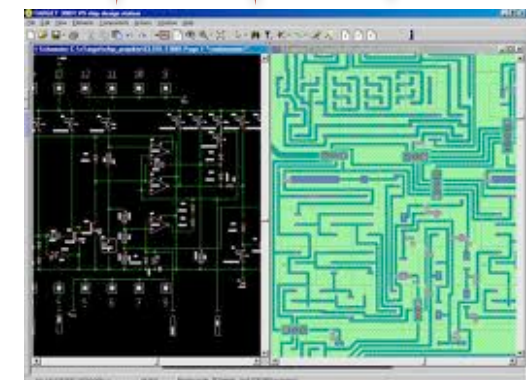
pthreads



multi C

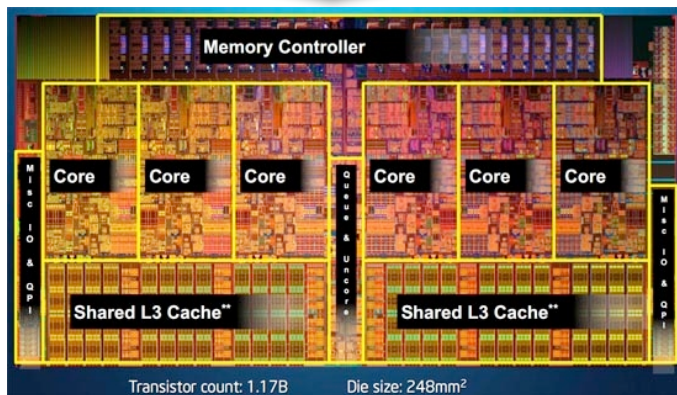


OpenCL

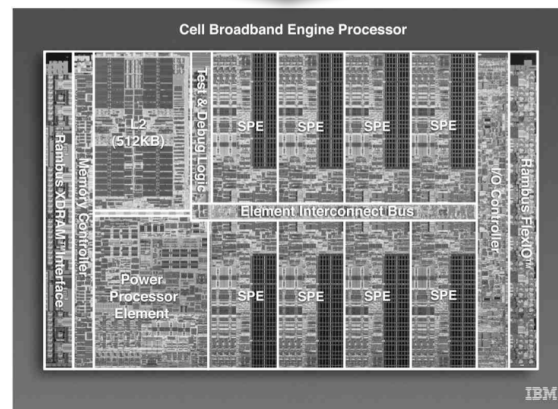


bitfile

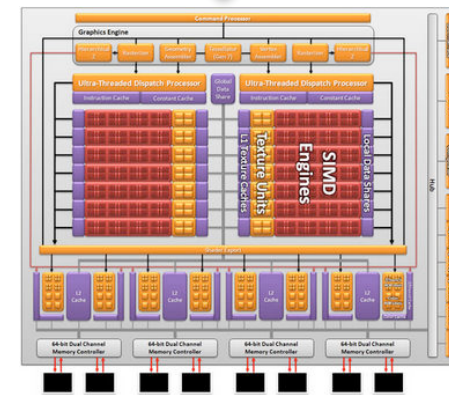
Legacy Program



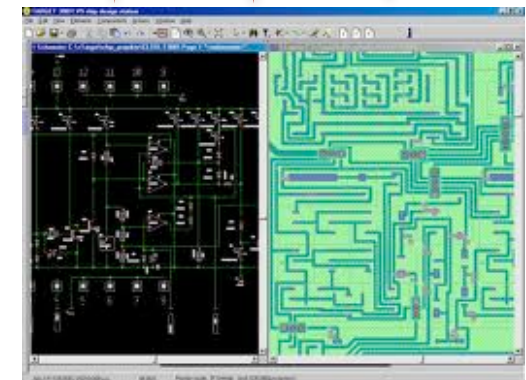
pthread



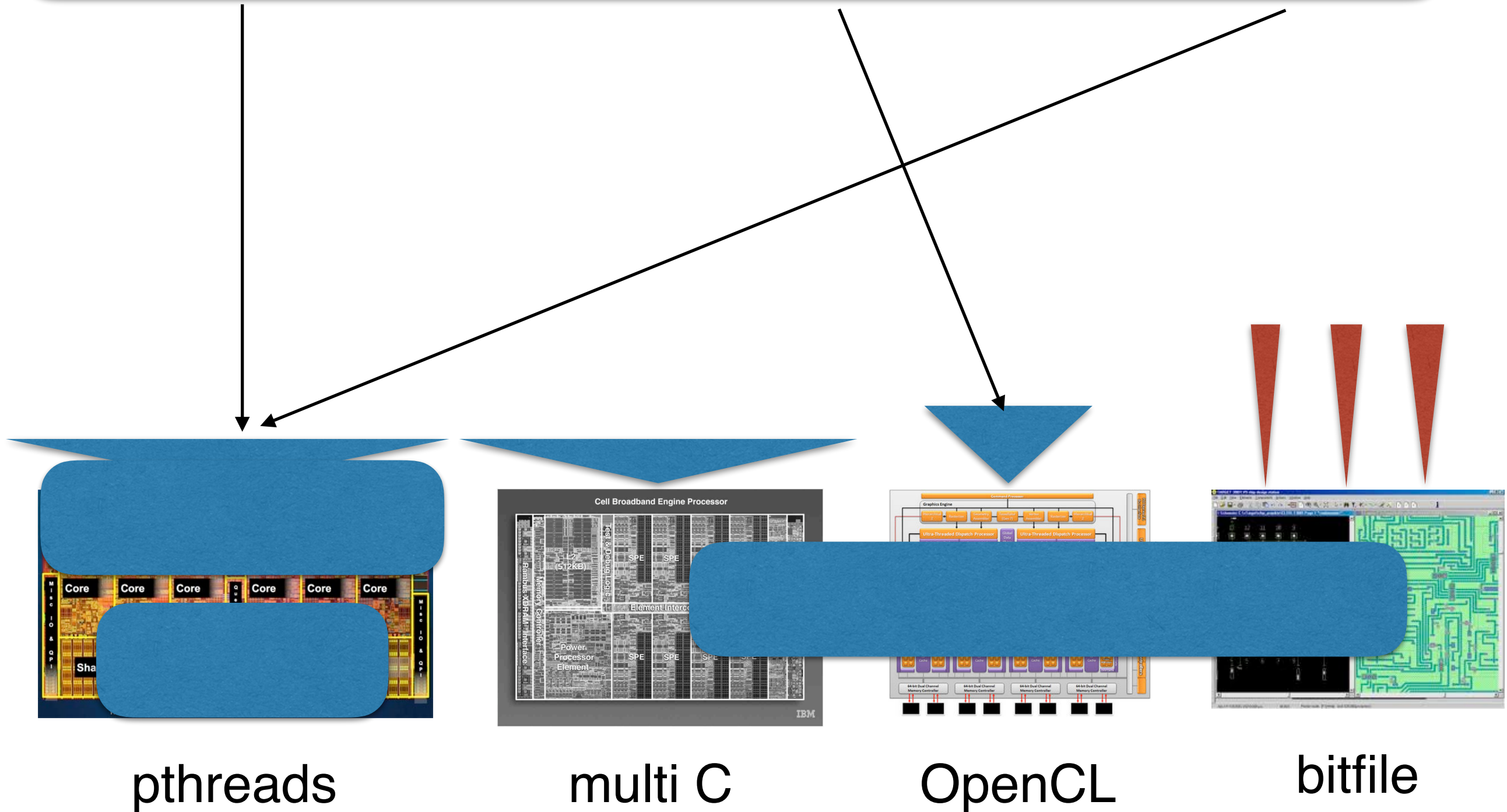
multi C



OpenCL

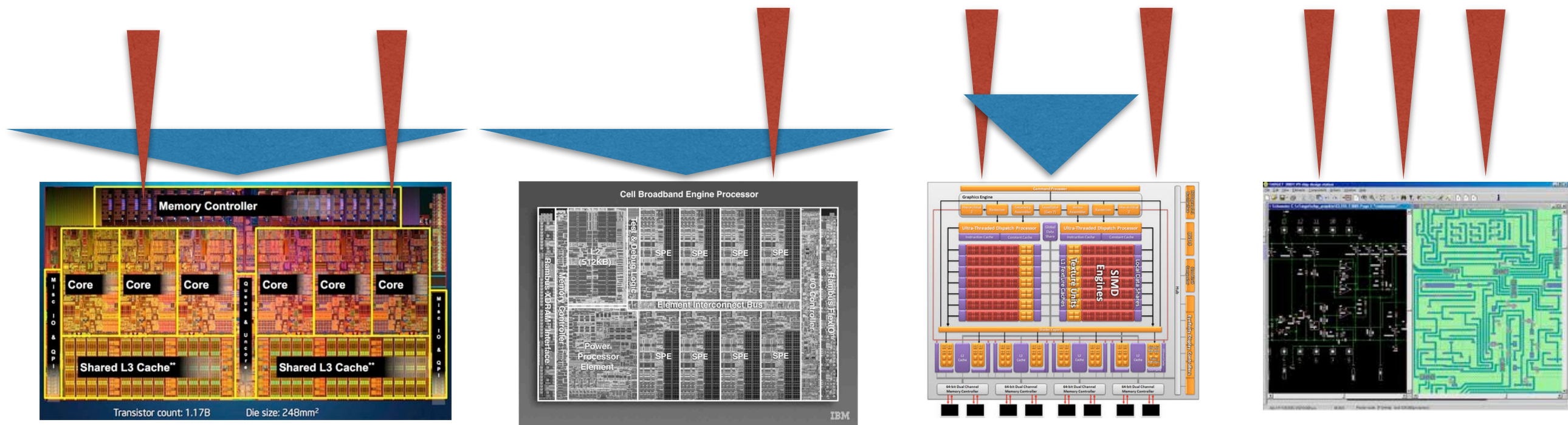


Legacy Program



Legacy Program

DSL/ Library/ API



Polly TBB
BLAS

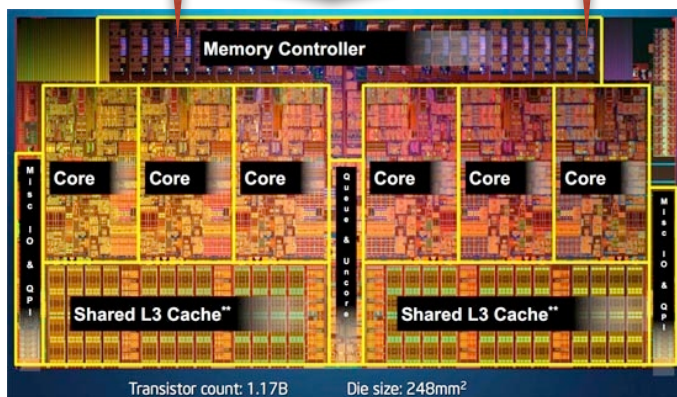
Milk
Halide

PolyACC Lift
OpenGL

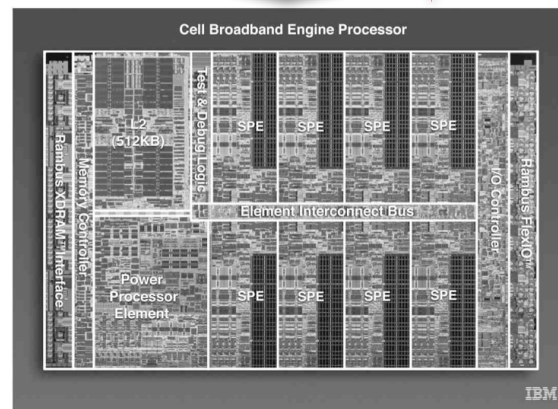
fir fft

Legacy Program

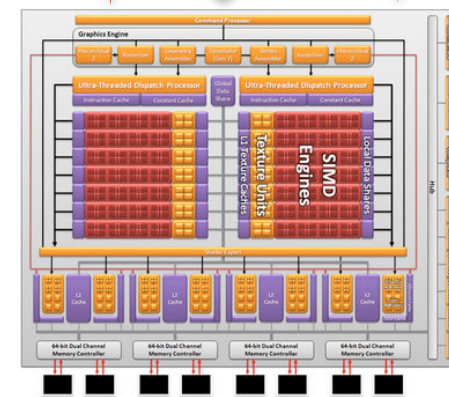
DSL/ Library/ API



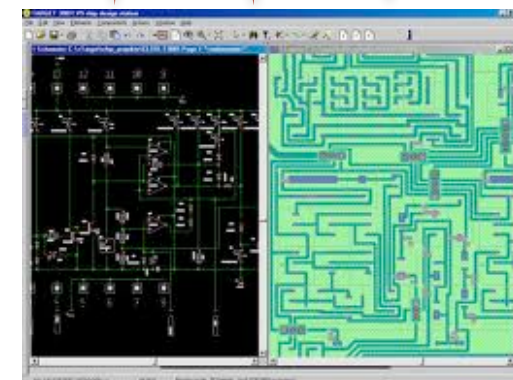
Polly TBB
BLAS



Milk
Halide

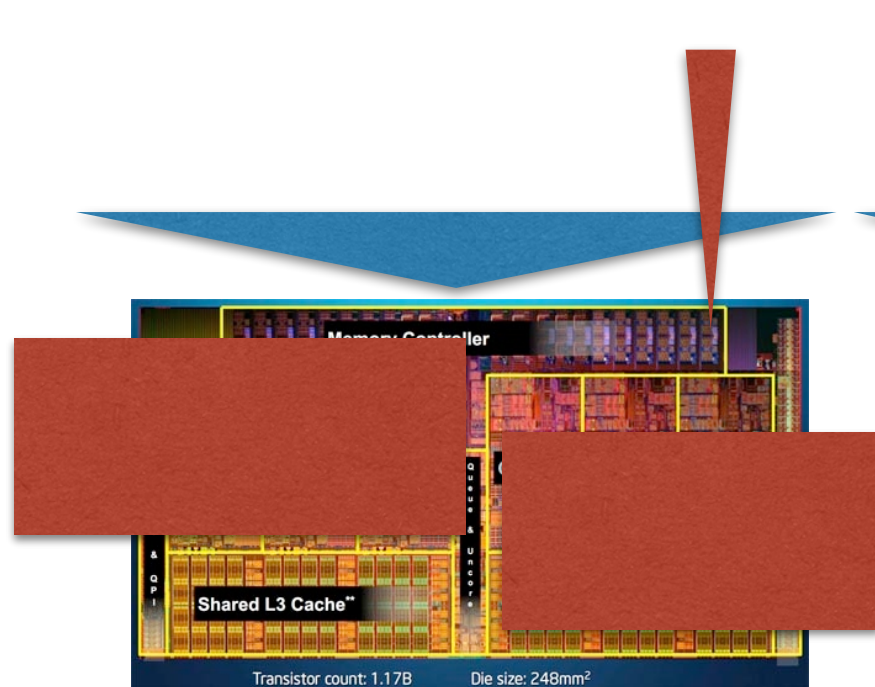


PolyACC Lift
OpenGL

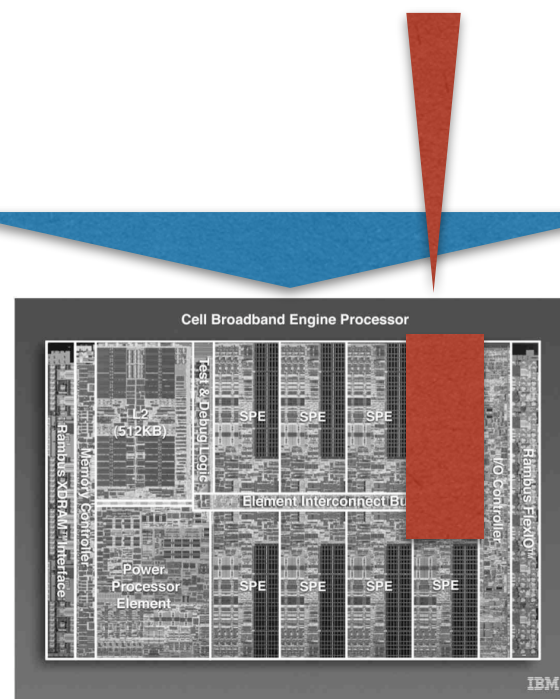


fir fft

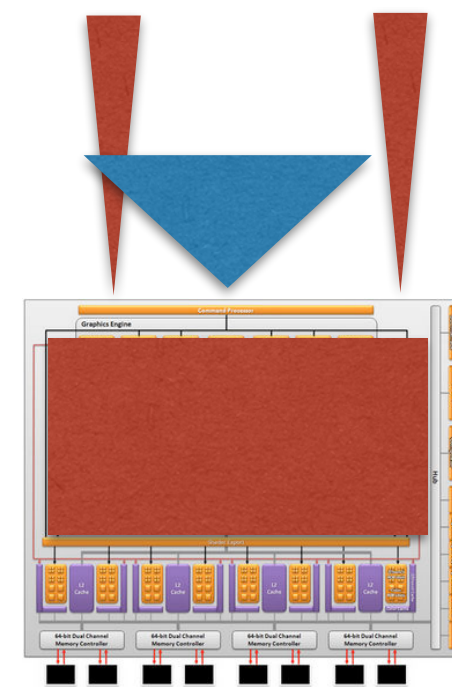
Legacy Program



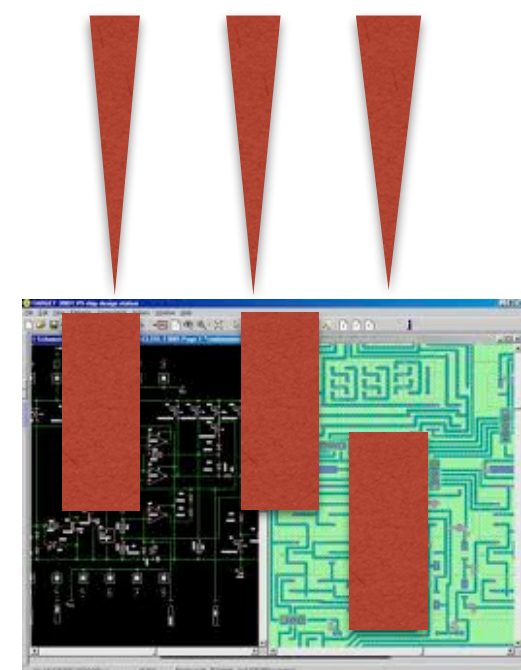
Polly TBB
BLAS



Milk
Halide



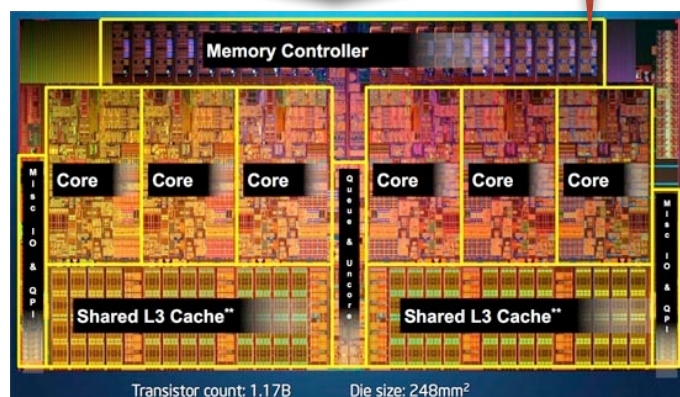
PolyACC Lift
OpenGL



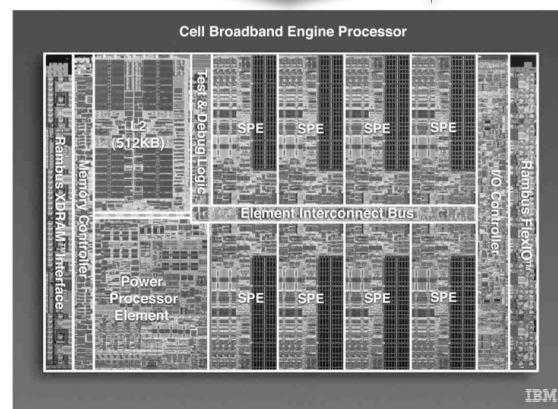
fir fft



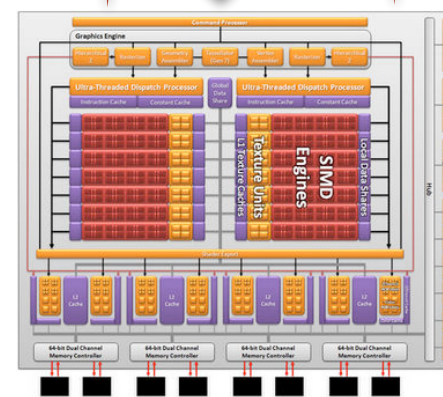
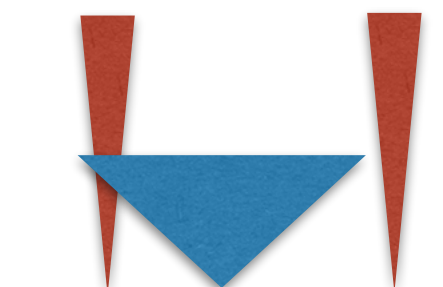
?



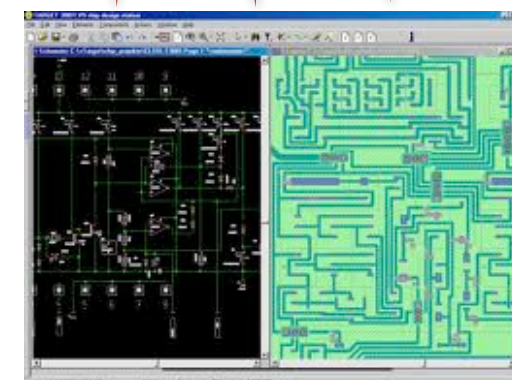
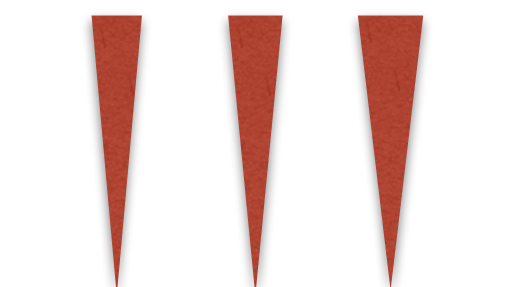
Polly TBB
BLAS



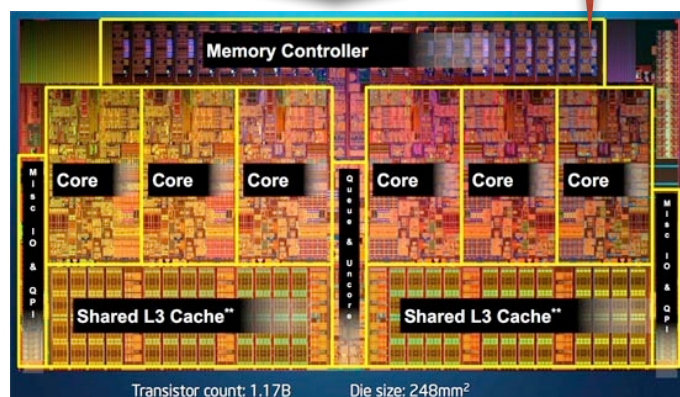
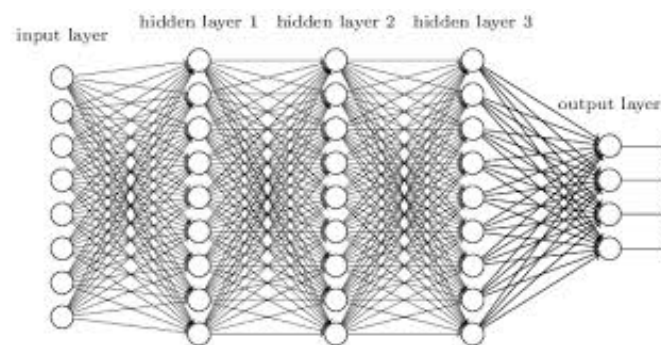
Milk
Halide



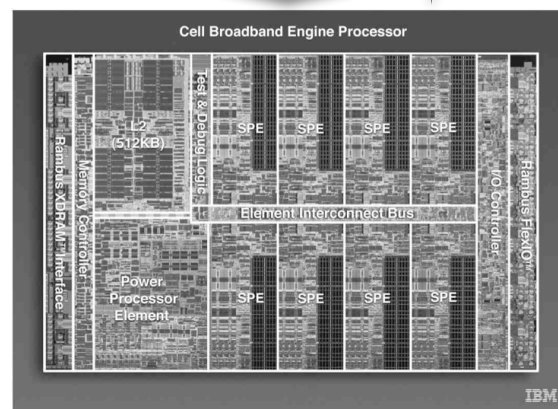
PolyACC Lift
OpenGL



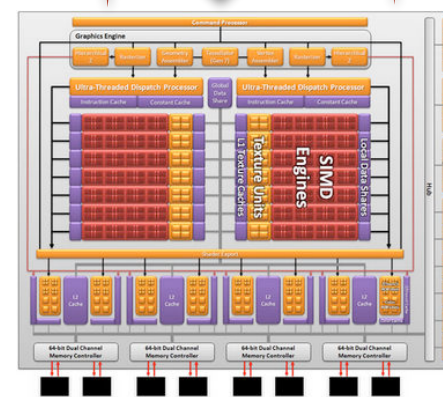
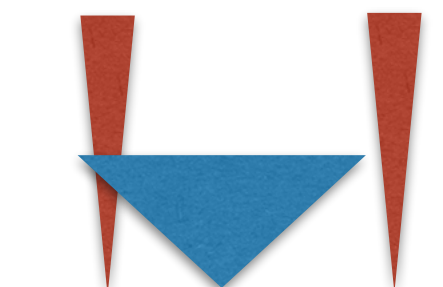
fir fft



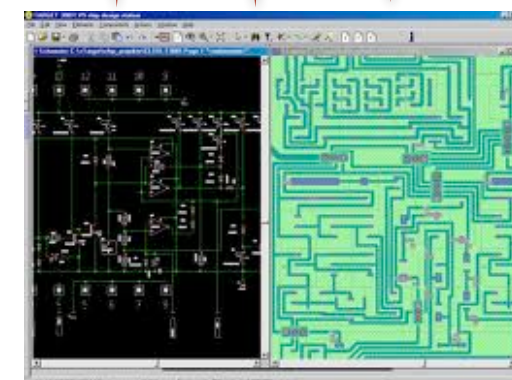
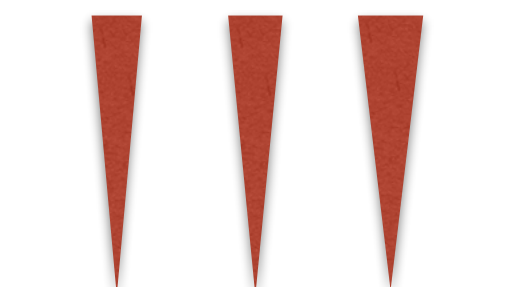
Polly TBB
BLAS



Milk
Halide

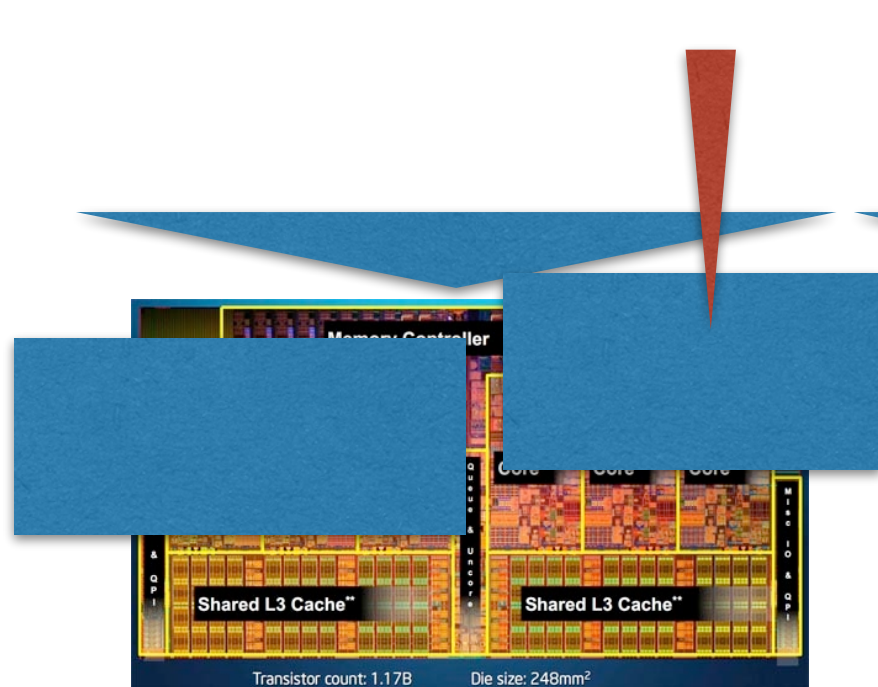
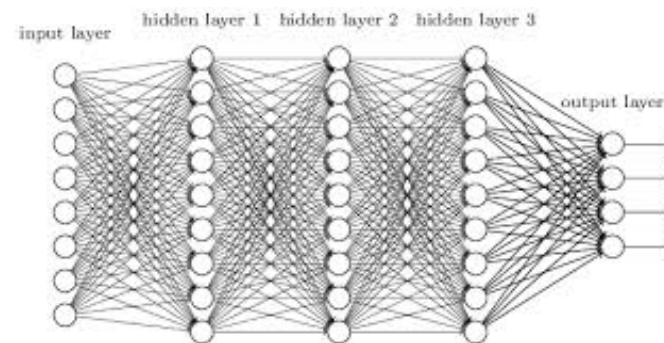


PolyACC Lift
OpenGL

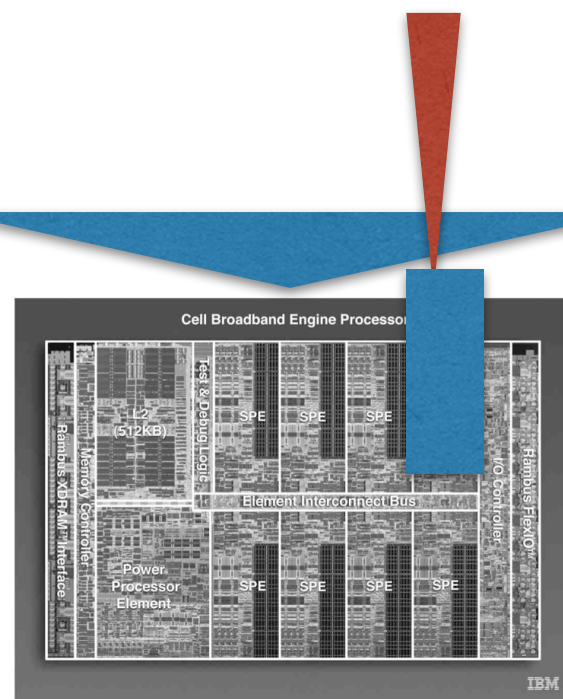


fir fft

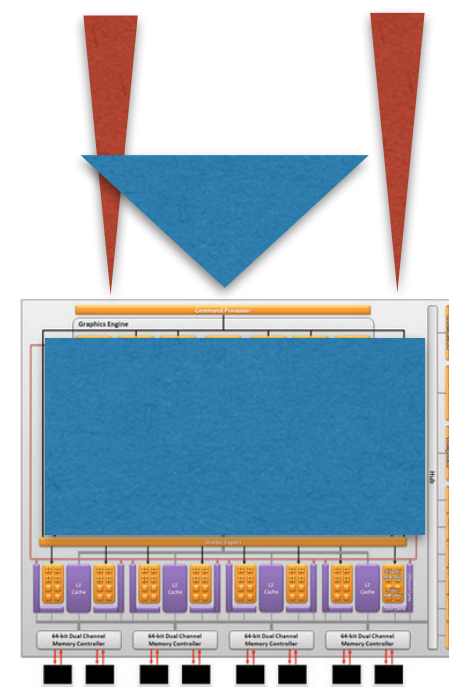
Legacy Program



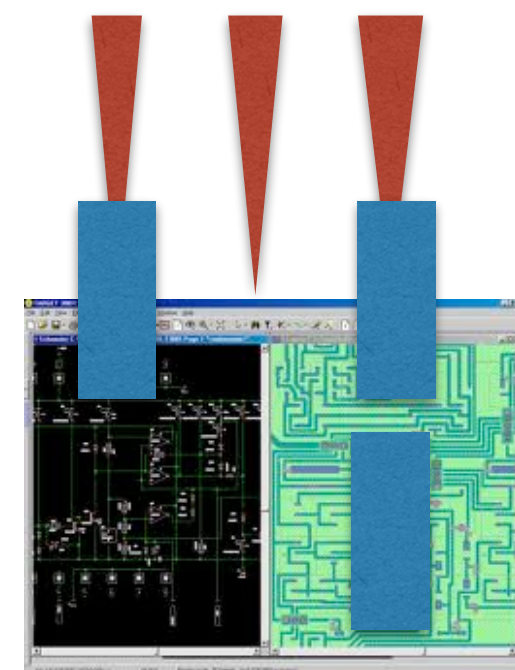
Polly TBB
BLAS



Milk
Halide



PolyACC Lift
OpenGL

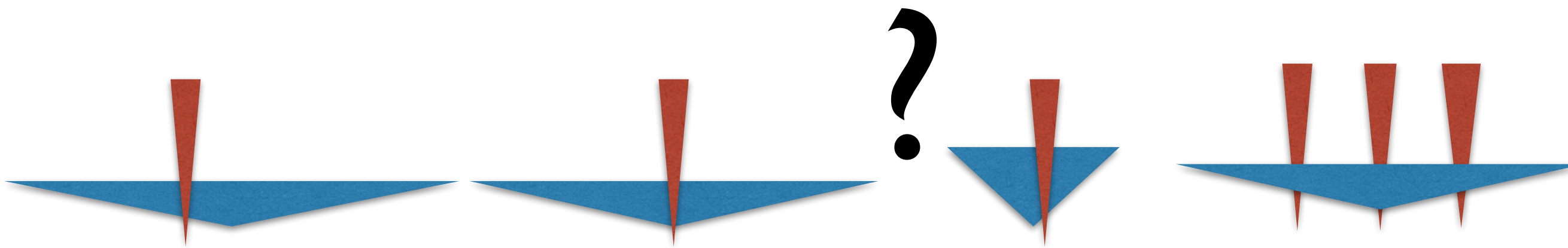


fir fft

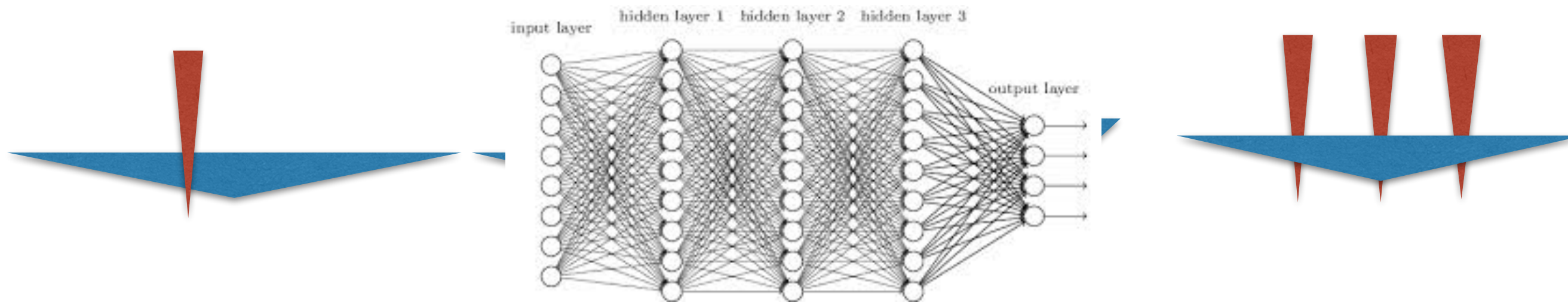


Space of Interesting Programs

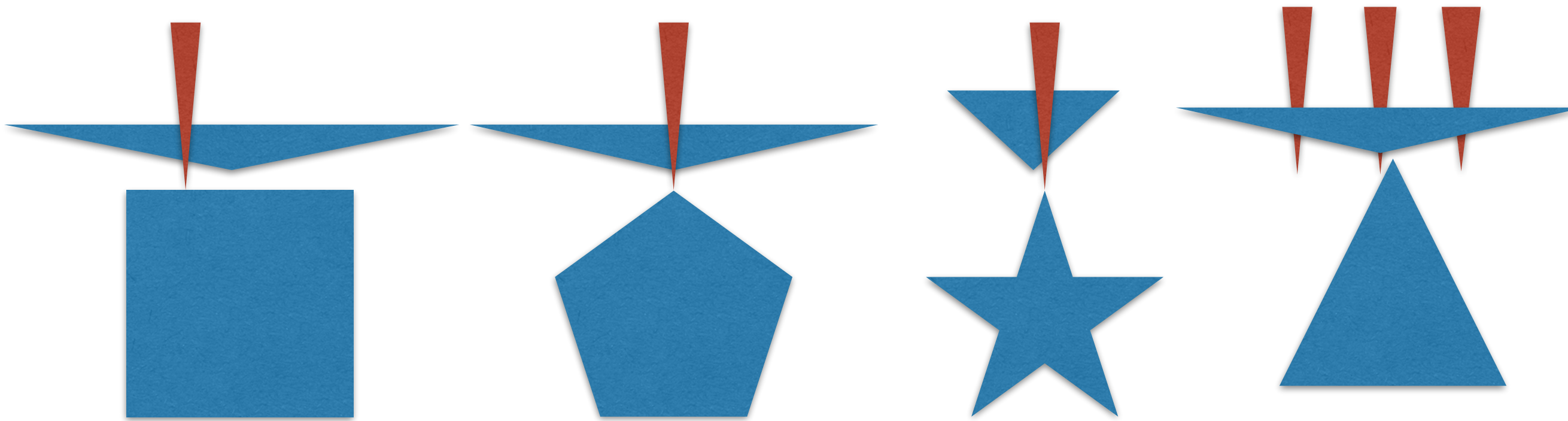
Space of Interesting Programs



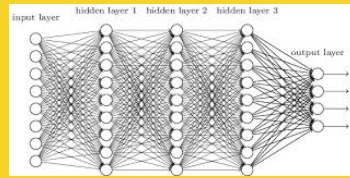
Space of Interesting Programs



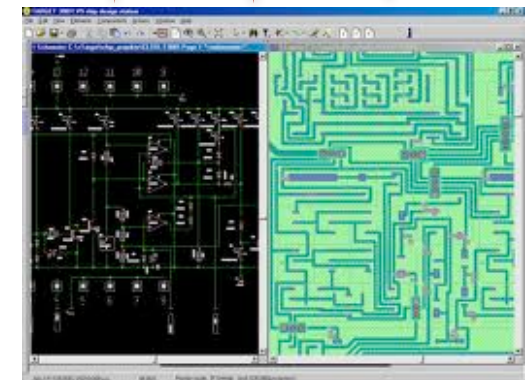
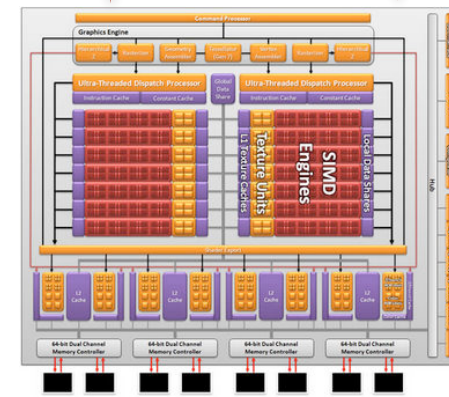
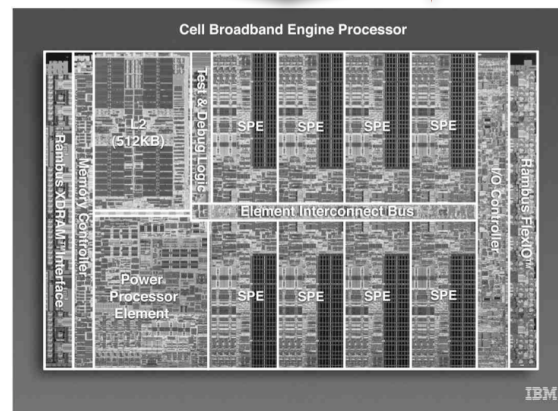
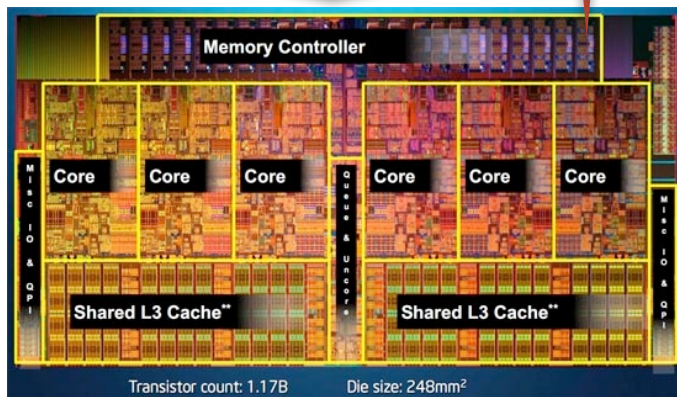
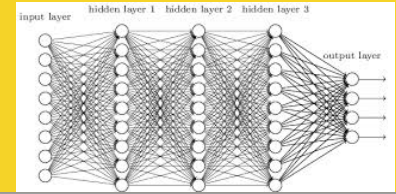
Space of Interesting Programs



... much later

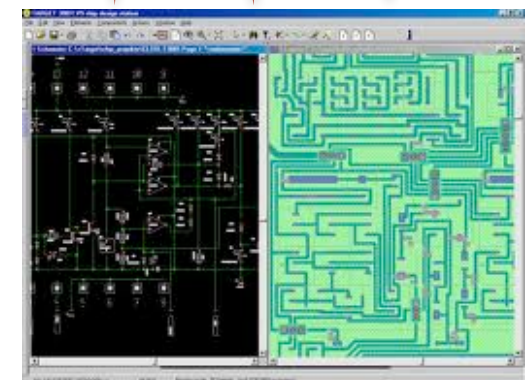
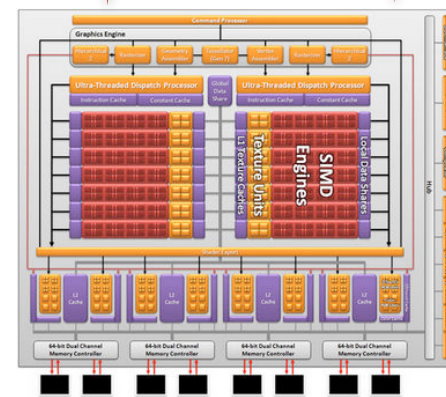
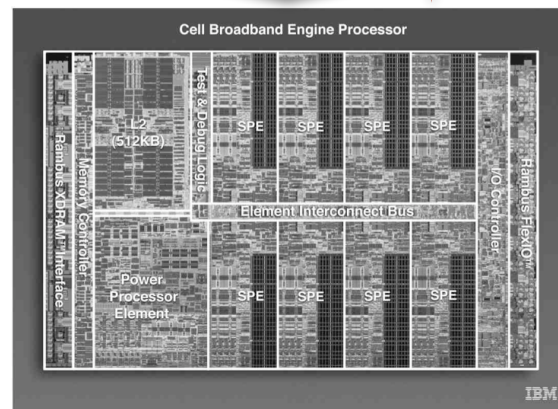
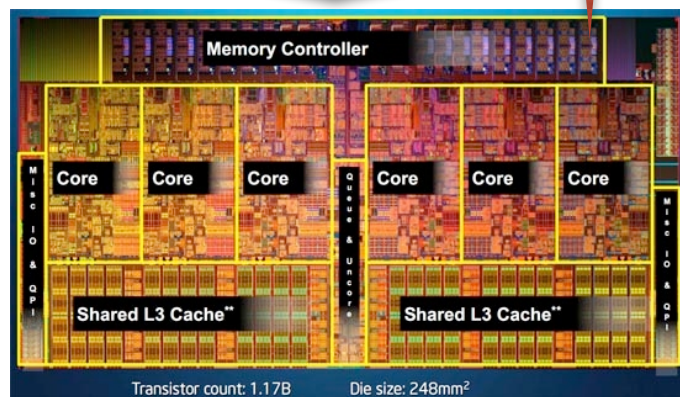
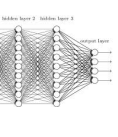
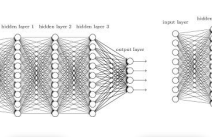
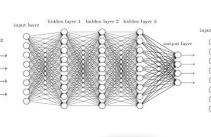
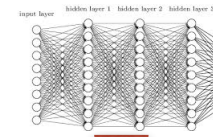
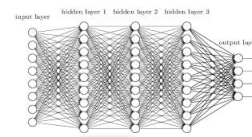
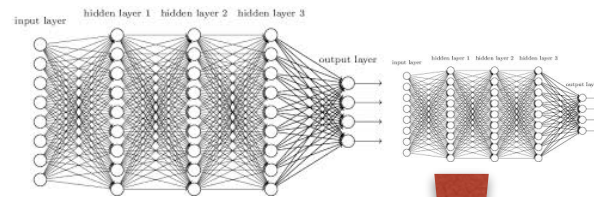
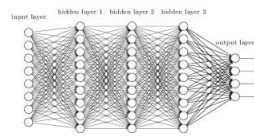
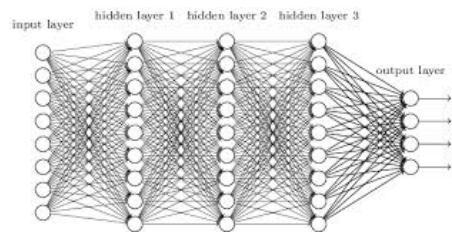
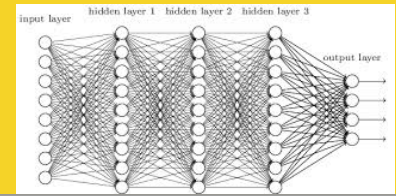
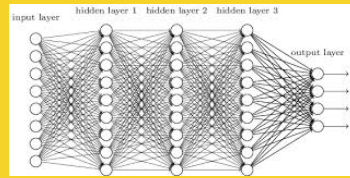


Legacy Program



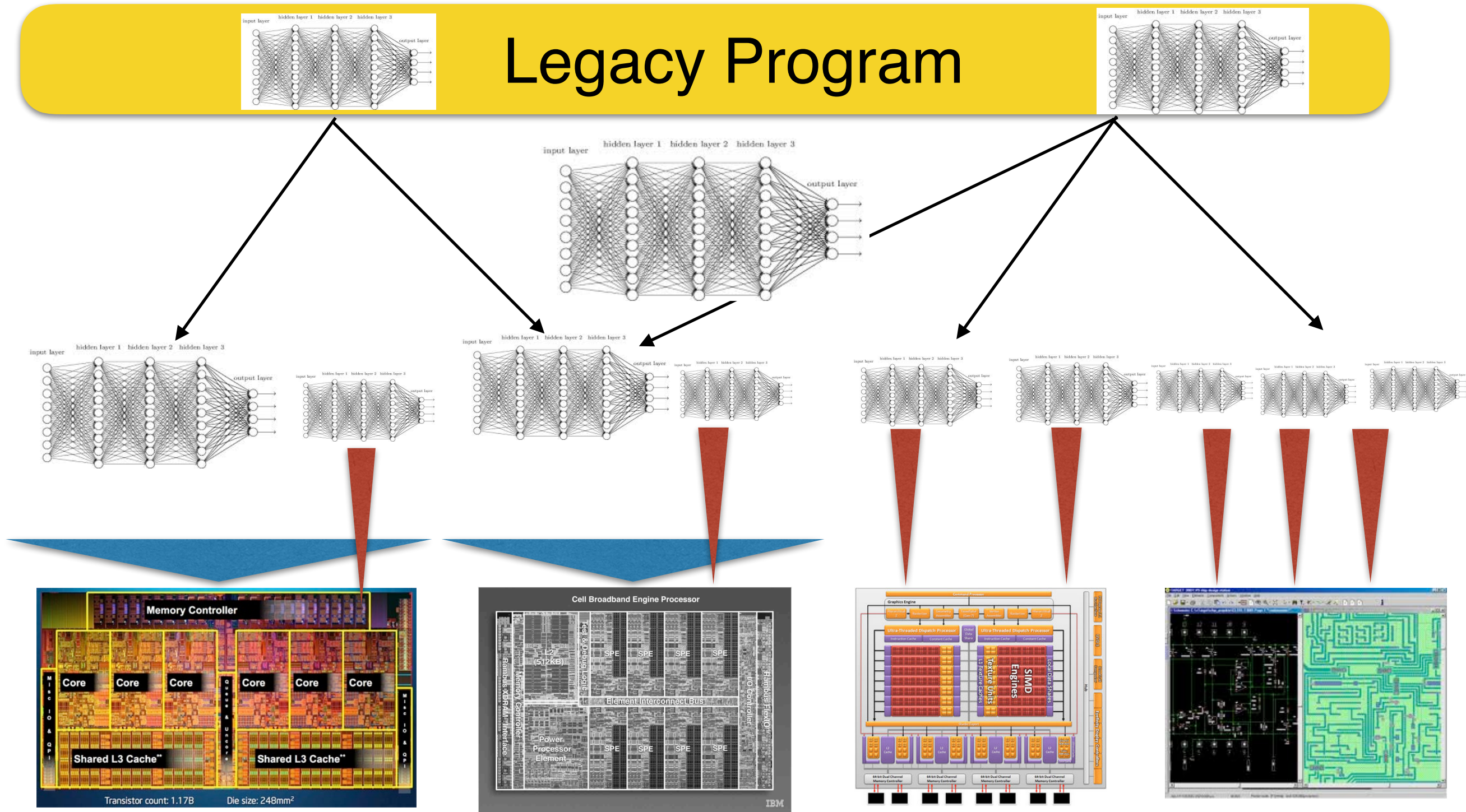
... much later

Legacy Program



... much much later

Legacy Program



Idioms and constraints

[ASPLOS18] **Automatic matching of legacy code to heterogeneous APIs: An idiomatic approach**

Libraries and DSLs are the new API/ISA

Detect code structures (idioms) that match APIs

Idioms:

- Dense linear algebra,
- Sparse Mv,
- Stencils,
- Reductions
- Histograms

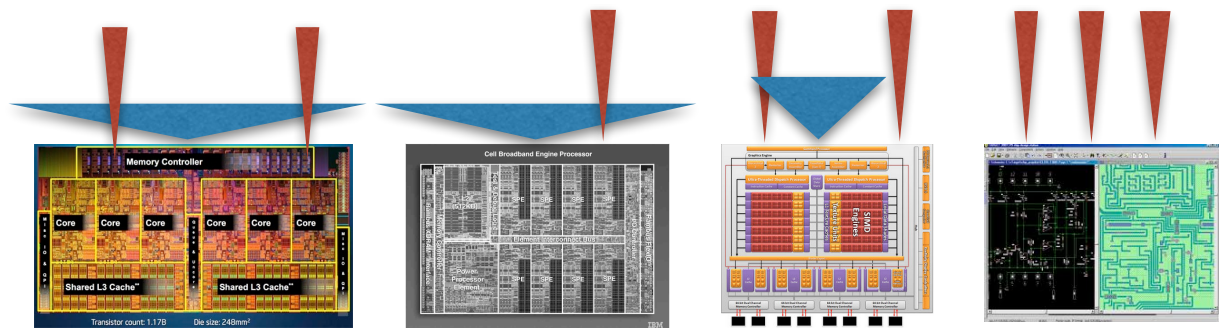


APIs:

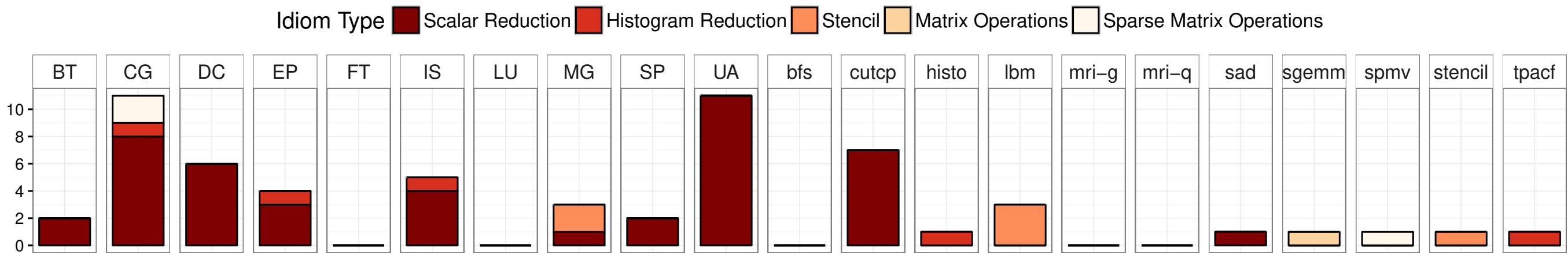
- cuBLAS, cIBLAS
- cuSparse, clSparse
- Halide, Lift

Platform

- AMD APU: multicore (+Radeon) (+NVIDIA Titan)

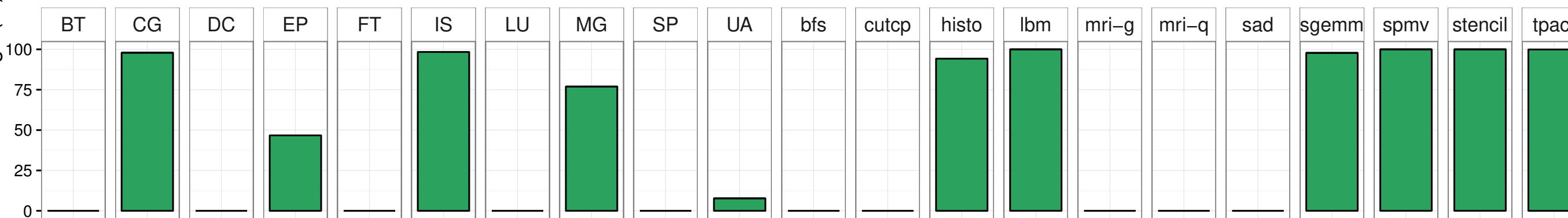


Evidence?



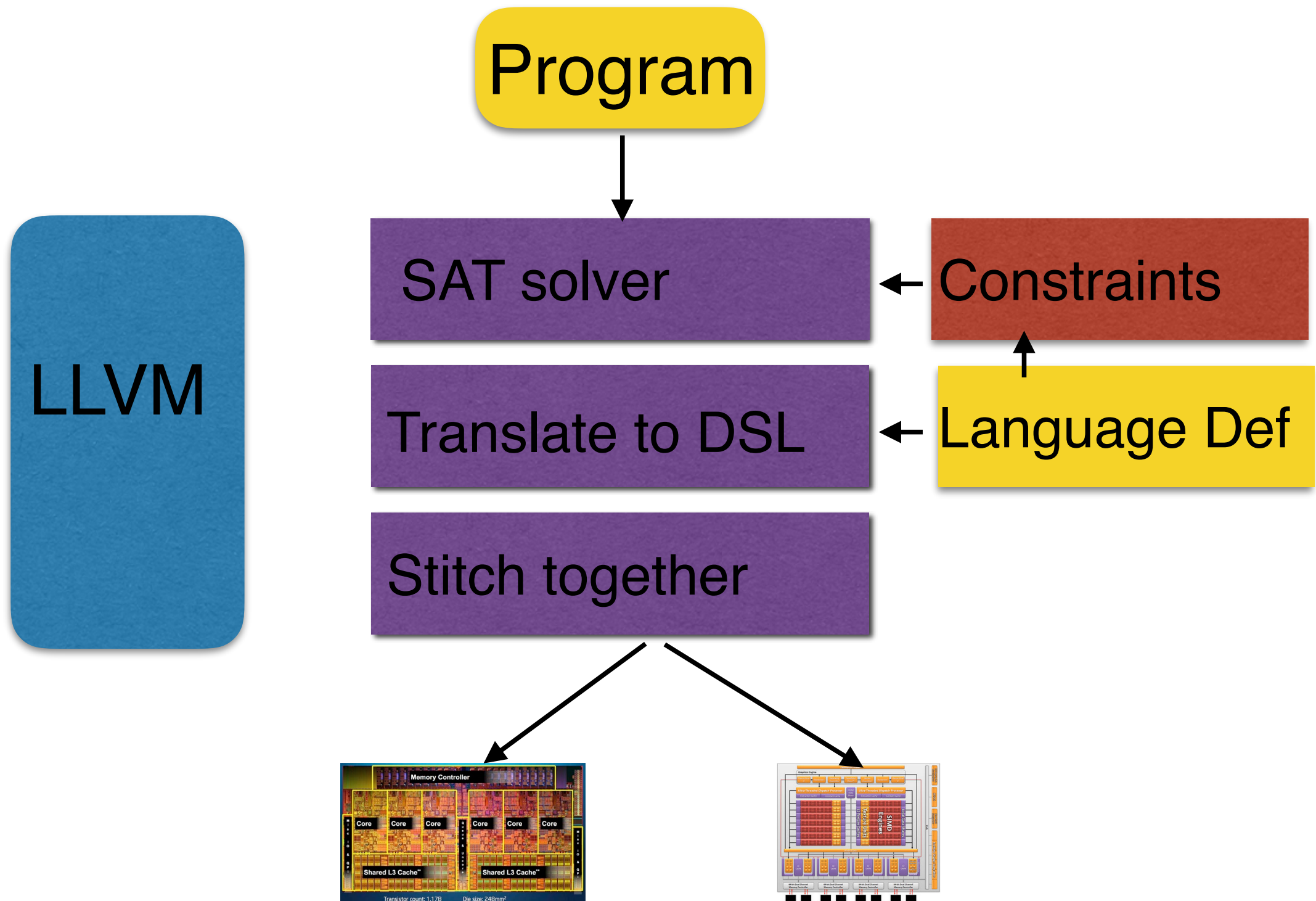
How many?

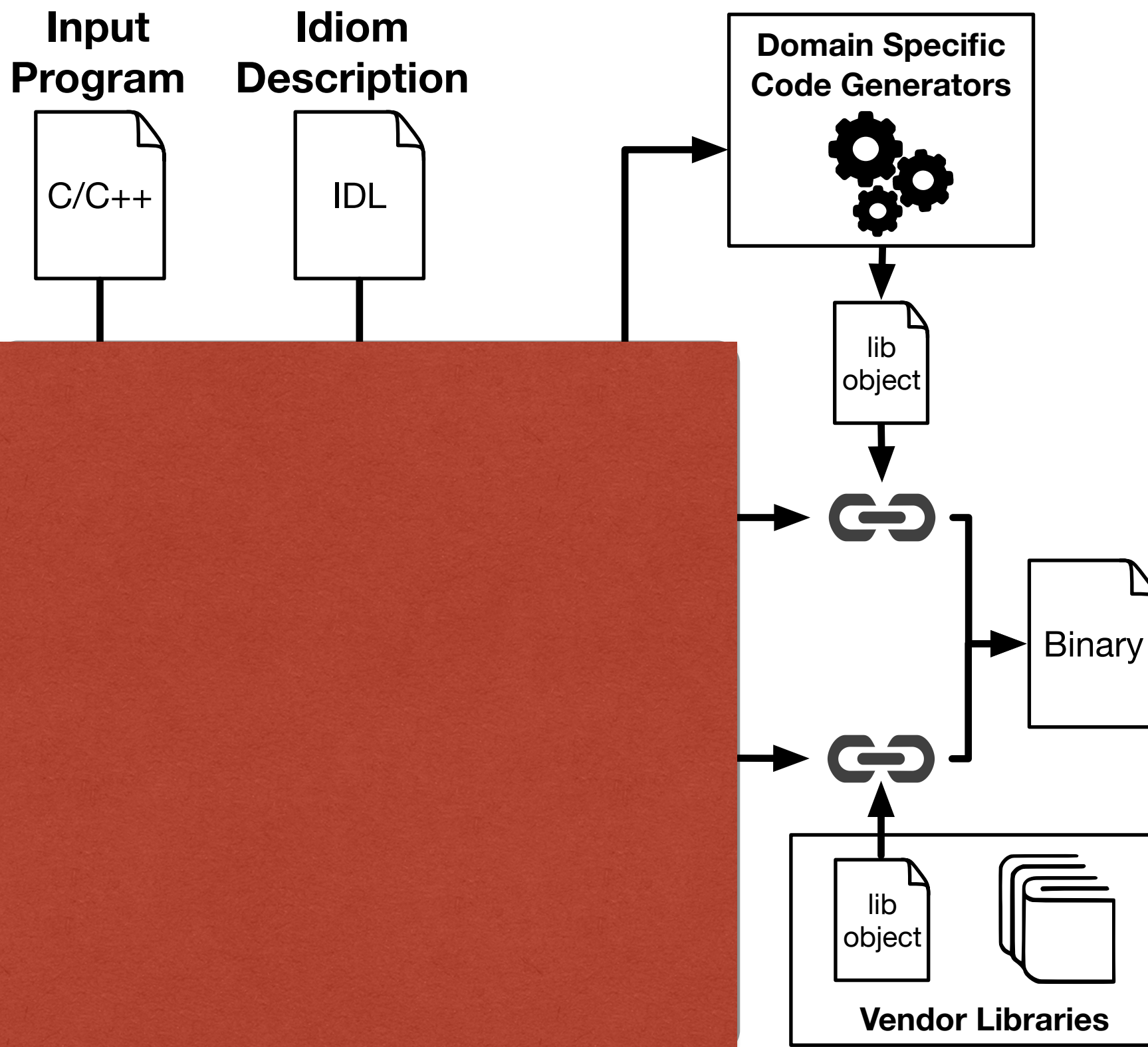
NAS PB +Parboil

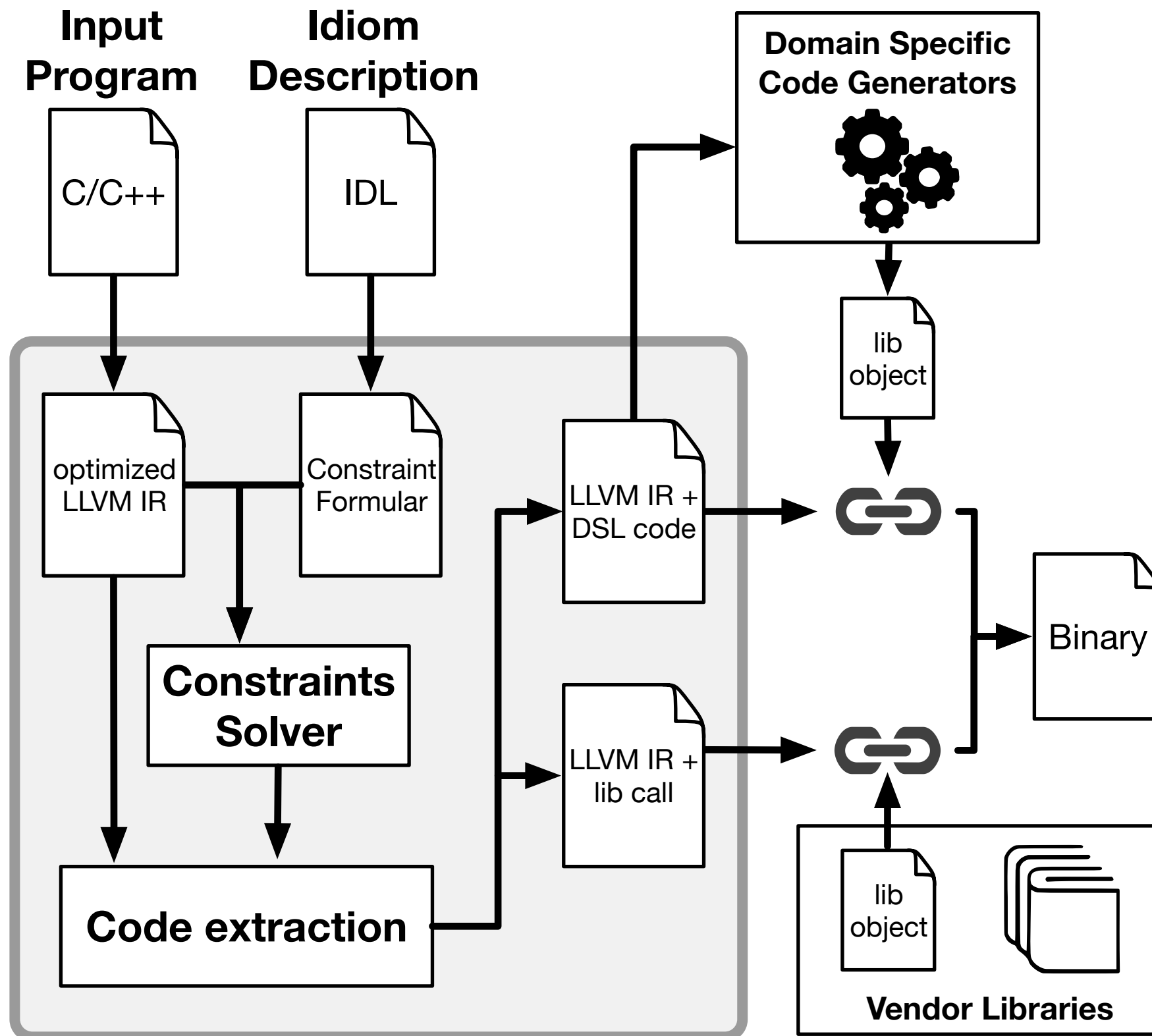


Coverage?

NAS PB +Parboil







$$(x * y) + (x * z) = x * (y + z)$$

Constraint FactorizationOpportunity

```
( {sum} is add instruction and
  {left_addend} is first argument of {sum} and
  {left_addend} is mul instruction and
  {right_addend} is second augment of {sum} and
  {right_addend} is mul instruction and
  ( {factor} is first argument of {left_addend} or
    {factor} is second argument of {left_addend}) and
  ( {factor} is first argument of {right_addend} or
    {factor} is second argument of {right_addend}))
```

End

```
int example(int a, int b, int c) {
    int d = a;
    return (a*b) + (c*d);
}
```

```
define i32 @example(i32 %a, i32 %b, i32 %c) {
    %1 = mul i32 %a, %b
    %2 = mul i32 %c, %a
    %3 = add i32 %1, %2
    ret i32 %3
}
```

```
1 { "sum" : %3,
2   "left_addend" : %1,
3   "right_addend" : %2,
4   "factor" : %a }
```



```

for (j = 0; j < m; j++) {
    d = 0.0;
    for (k = rowstr[j]; k < rowstr[j+1]; k++)
        d = d + a[k]*z[colidx[k]];
    r[j] = d; }

```



```

Constraint SPMV
( inherits For and
  inherits VectorStore
    with {iterator} as {idx}
    and {begin} as {begin} at {output} and
  inherits ReadRange
    with {iterator} as {idx}
    and {inner.iter_begin} as {range_begin}
    and {inner.iter_end} as {range_end} and
  inherits For at {inner} and
  inherits VectorRead
    with {inner.iterator} as {idx}
    and {begin} as {begin} at {idx_read} and
  inherits VectorRead
    with {idx_read.value} as {idx}
    and {begin} as {begin} at {indir_read} and
  inherits VectorRead
    with {inner.iterator} as {idx}
    and {begin} as {begin} at {seq_read} and
  inherits DotProductLoop
    with {inner} as {loop}
    and {indir_read.value} as {src1}
    and {seq_read.value} as {src2}
    and {output.address} as {update_address})
End

```



```

cusparseDcsrmmv(context,
    CUSPARSE_OPERATION_NON_TRANSPOSE, m, n,
    rowstr[m+1]-rowstr[0], &gpu_1, descr, gpu_a,
    gpu_rowstr, gpu_colidx, gpu_z, &gpu_0, gpu_r);

```

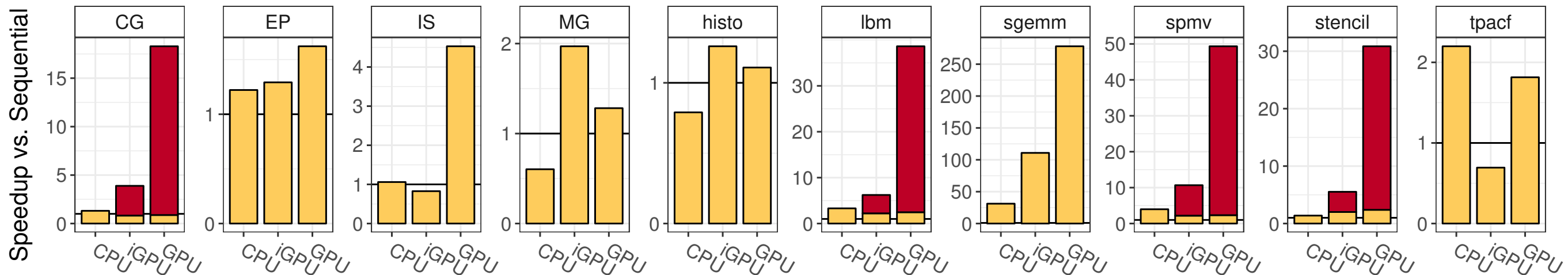
Results

	Scalar Reduction	Histogram Reduction	Stencil	Matrix Op.	Sparse Matrix Op.
Polly	3	—	5	—	—
ICC	28	—	—	—	—
IDL	45	5	6	1	3

NAS Parallel Benchmarks - sequential C code

Parboil Benchmarks - sequential C code

Results



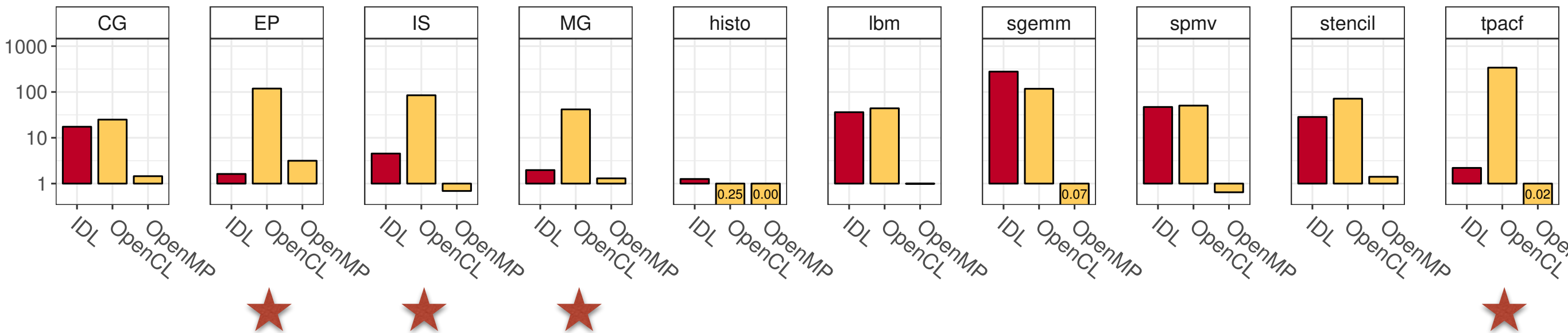
Speedup over sequential code 1.1x to 250x

Polly and ICC slow down - not shown

Automatically finds and exploits parallel idioms

- not attempted in parallelising compilers
- first to do this

Results



Speedup relative to (external) hand written code
- competitive in 5 cases

★ = user changed algorithm to create parallelism
- we can now detect this

Auto-discovery: Program Synthesis

Constraint GEMM

```
( inherits ForNest(N=3) and
  inherits MatrixStore
    with {iterator[0]} as {col}
    and {iterator[1]} as {row}
    and {begin} as {begin} at {output} and
    inherits MatrixRead
```

Constraint GEMM

```
( inherits ForNest(N=3) and
  inherits MatrixStore
    with {iterator[0]} as {col}
    and {iterator[1]} as {row}
    and {begin} as {begin} at {output} and
    inherits MatrixRead
```

Constraints

Examples

```
for (int mm = 0; mm < m; ++mm) {
  for (int nn = 0; nn < n; ++nn) {
    float c = 0.0f;
    for (int i = 0; i < k; ++i) {
      float a = A[mm + i * lda];
      float b = B[nn + i * ldb];
      c += a * b;
    }
    C[mm+nn*ldc] = C[mm+nn*ldc] * beta + alpha * c;
  }
}
```

```
for(int i = 0; i < 1000; i++)
  for(int j = 0; j < 1000; j++) {
    M3[i][j] = 0.0f;
    for(int k = 0; k < 1000; k++)
      M3[i][j] += M1[i][k] * M2[k][j];
  }
```

Constraints

Interrogate

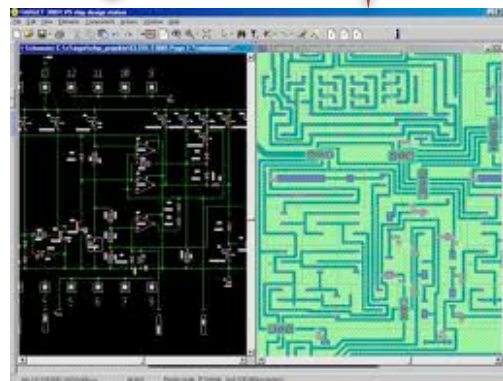


Hardware synthesis meets software synthesis

Legacy Program

SDH

HDS



Summary

Heterogeneity: avoiding abstraction tax

- Rethinking the hardware/software API

Automatically match software to any API

- Use this to design future hardware

IDL: matching code to libraries/DSLs

- Outperforms existing approaches

Automatically learn hardware behaviour and match to code

Rethinking the Hardware/Software Contract

Michael O'Boyle
University of Edinburgh

icsa | Institute for Computing
Systems Architecture



Heterogeneous Thinking

Michael O'Boyle
University of Edinburgh

icsa

Institute for Computing
Systems Architecture

