C2HLSC

Leveraging LLMs to refactor C code into HLS-compatible C

Luca Collini, **Andrew Hennessee**, Ramesh Karri, Siddharth Garg

NYU | TANDON SCHOOL OF ENGINEERING

What do system architects do?

What do system architects do?

1 Profile application

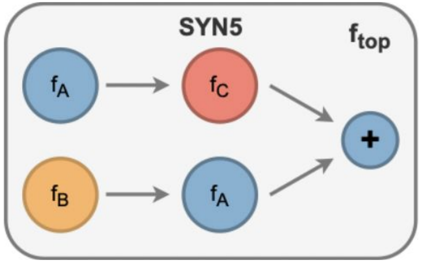
$f_C(f_A) + f_A(f_B)$

What do system architects do?

1 Profile application

$f_C(f_A) + f_A(f_B)$

2 Construct DFG

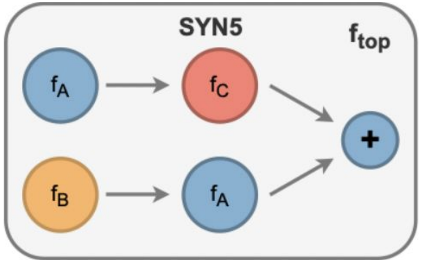


What do system architects do?

1 Profile application

$f_C(f_A) + f_A(f_B)$

2 Construct DFG



3 Model latency

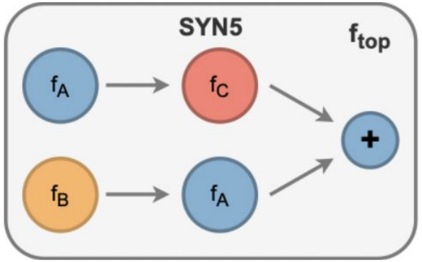
$$f_{\text{top}} + \max(f_A + f_C, f_B + f_A)$$

What do system architects do?

1 Profile application

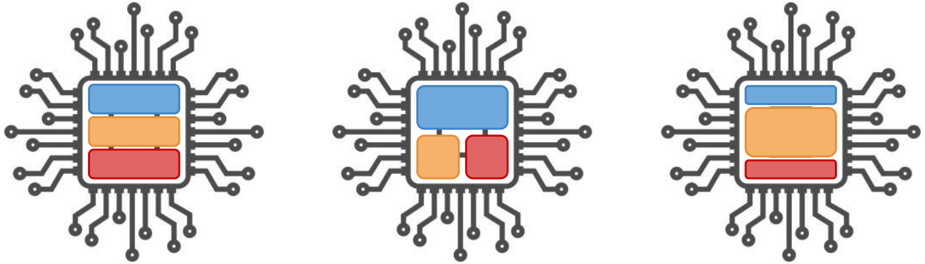
$f_C(f_A) + f_A(f_B)$

2 Construct DFG



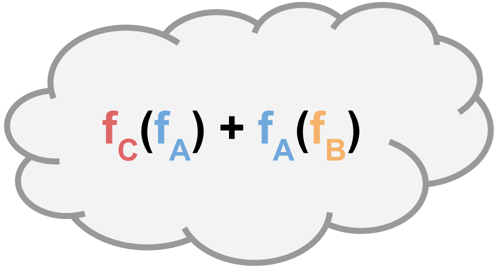
3 Model latency $f_{\text{top}} + \max(f_A + f_C, f_B + f_A)$

4 DSE to map kernels to hardware



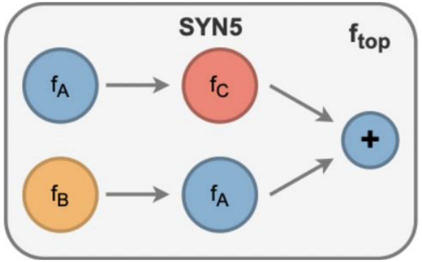
Can LLMs replace system architects?

1 Profile application



$f_C(f_A) + f_A(f_B)$

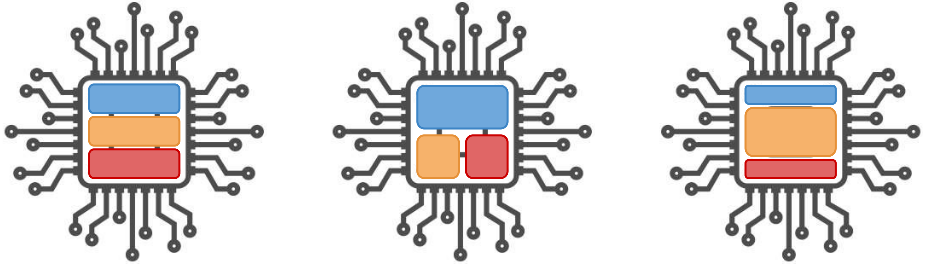
2 Construct DFG



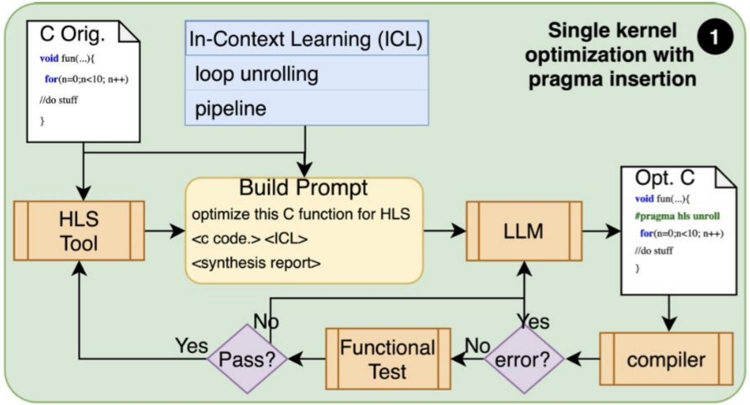
3 Model latency

$$f_{\text{top}} + \max(f_A + f_C, f_B + f_A)$$

4 DSE to map kernels to hardware



Overview of proposed flow



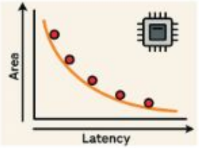
```
void fun(...){
  for(n=0;n<10; n++)
//do stuff
}
```

optimize this C function for HLS
<c code.> <ICL>
<synthesis report>

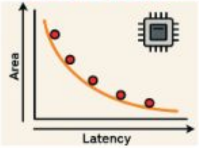
```
void fun(...){
#pragma hls unroll
  for(n=0;n<10; n++)
//do stuff
}
```

Single Kernel Optimization

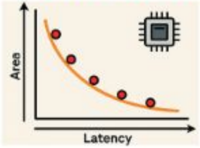
Kernel 1

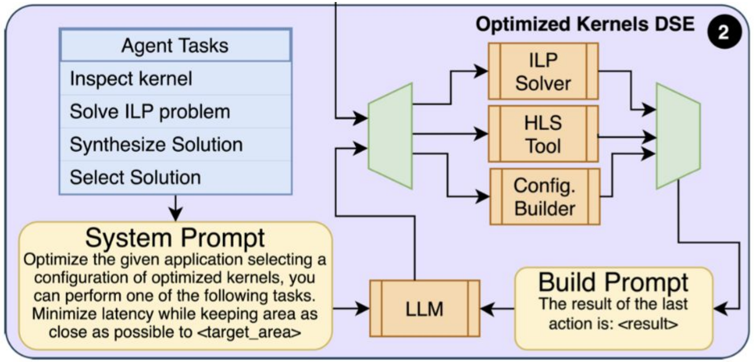


Kernel 2



Kernel *n*





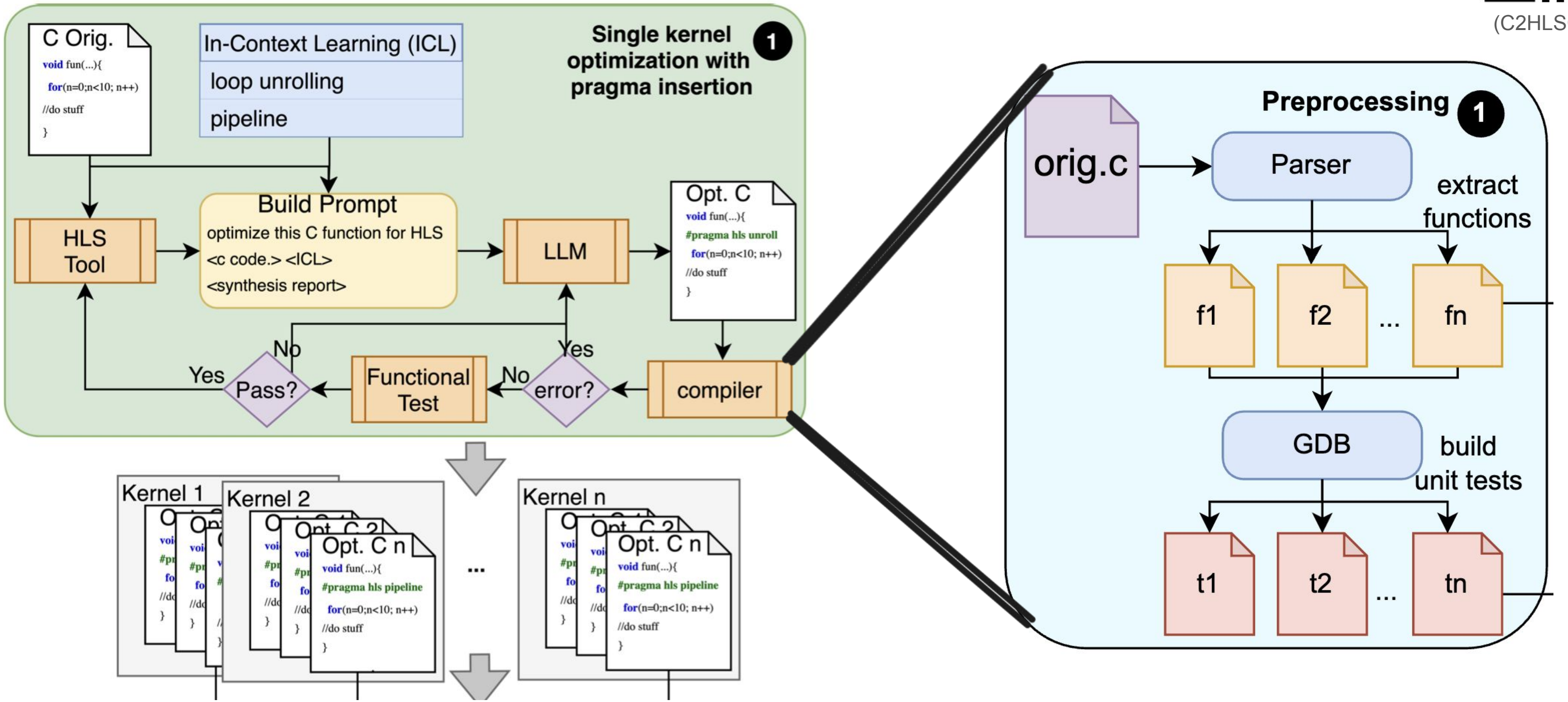
Optimize the given application selecting a configuration of optimized kernels, you can perform one of the following tasks. Minimize latency while keeping area as close as possible to <target_area>

The result of the last action is: <result>

Full System Composition

Single kernel optimization via pragma insertion

(C2HLSC, TODAES)



C Orig.:

```
void fun(...){
  for(n=0;n<10; n++)
 //do stuff
 }
```

Opt. C:

```
void fun(...){
#pragma hls unroll
  for(n=0;n<10; n++)
//do stuff
}
```

Opt. C n:

```
void fun(...){
#pragma hls pipeline
  for(n=0;n<10; n++)
//do stuff
}
```

Design points for DSE

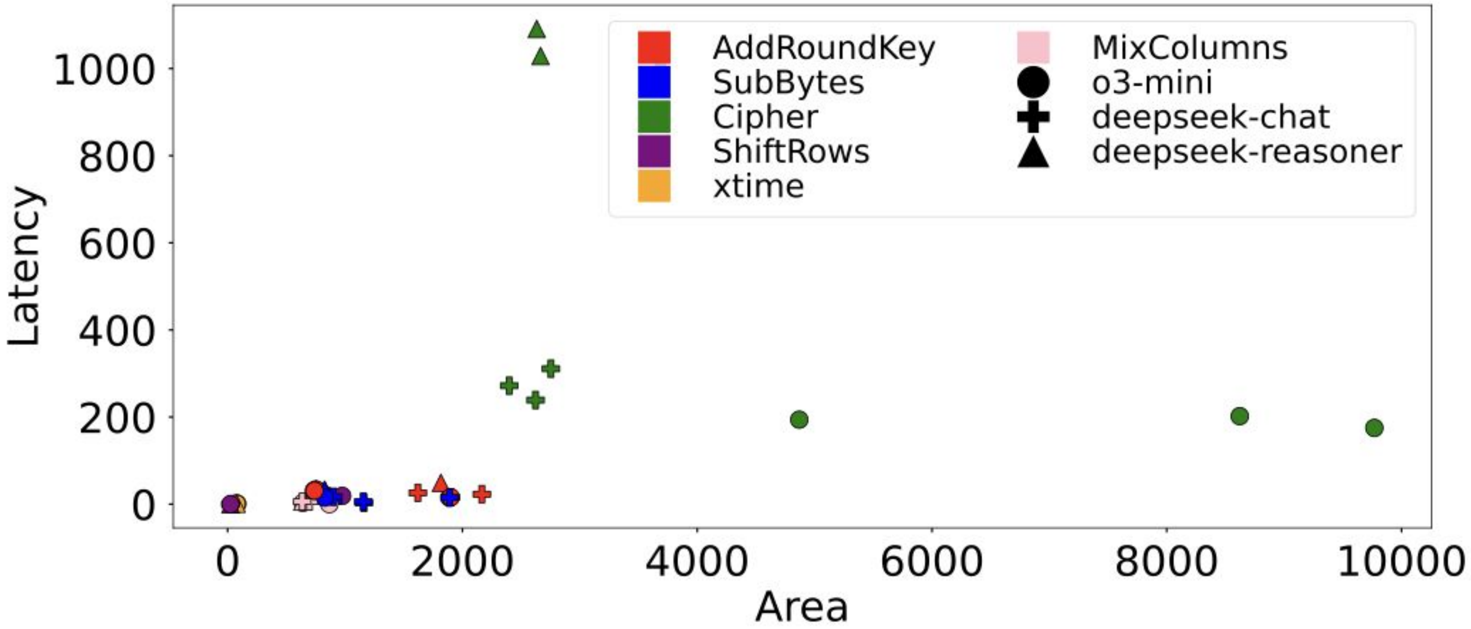
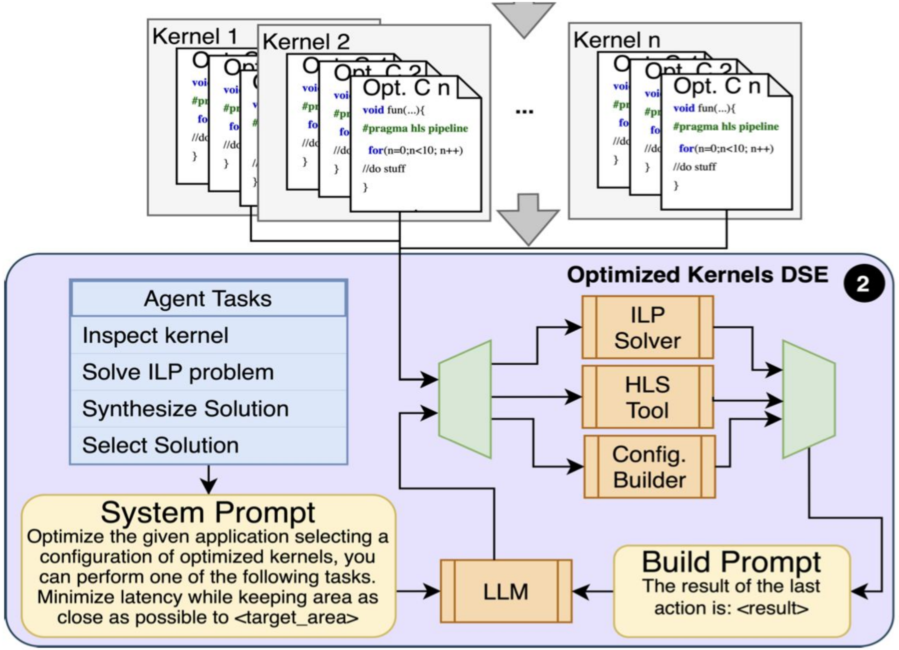


Fig. 4: Solutions for AES sub-kernels for each model.

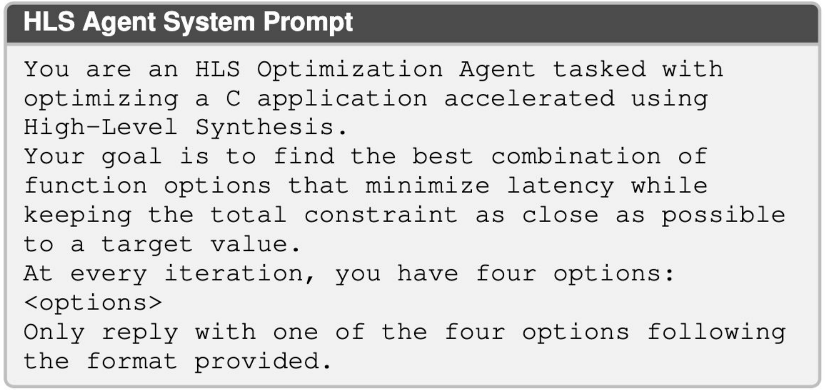
DSE of optimized kernels



```
void fun(...){
#pragma hls pipeline
  for(n=0;n<10; n++)
//do stuff
}
```

Optimize the given application selecting a configuration of optimized kernels, you can perform one of the following tasks. Minimize latency while keeping area as close as possible to <target_area>

The result of the last action is: <result>



HLS Agent System Prompt

```
You are an HLS Optimization Agent tasked with
optimizing a C application accelerated using
High-Level Synthesis.
Your goal is to find the best combination of
function options that minimize latency while
keeping the total constraint as close as possible
to a target value.
At every iteration, you have four options:
<options>
Only reply with one of the four options following
the format provided.
```

Thank you!

NYU | TANDON SCHOOL OF ENGINEERING