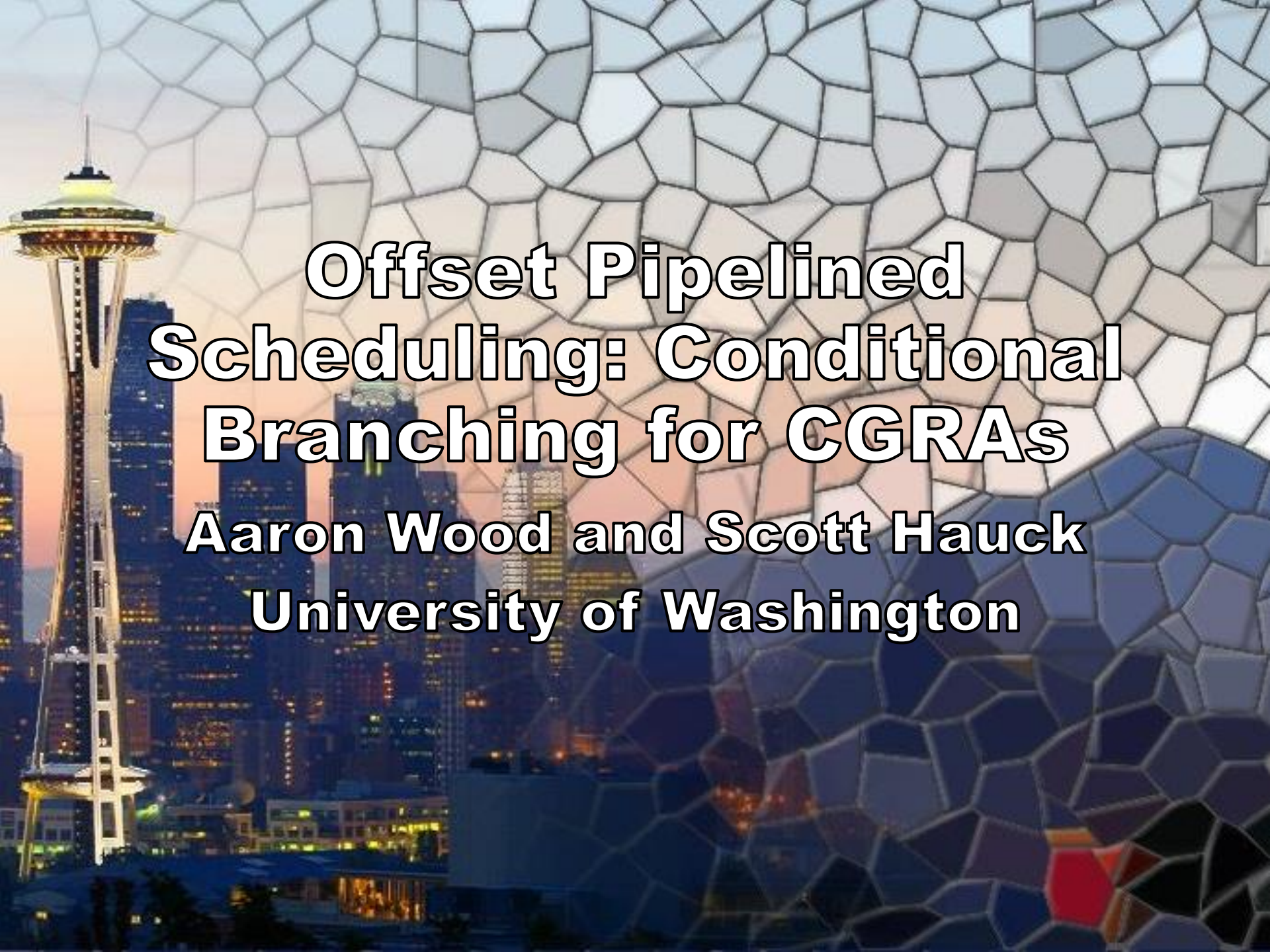


Offset Pipelined Scheduling: Conditional Branching for CGRAs

Aaron Wood and Scott Hauck
University of Washington

Motivating Example

```
while (1) {
   s = t;
   for (int i = 0; i < SIZE; i++) {
      x <? InputStream;
      s = s + x;
      a[i] = x;
   }
   for (int j = 0; j < SIZE; j++) {
      y = a[j];
      t = y ^ t;
      y = y * PRIME;
      y = y & 0x000000FF;
      y = BASIS ^ y;
      y = y / s;
      OutputStream <! y;
   }
}
```

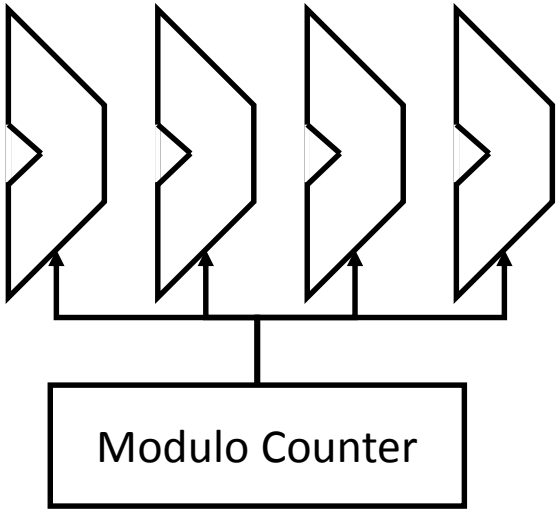
Modulo Schedule

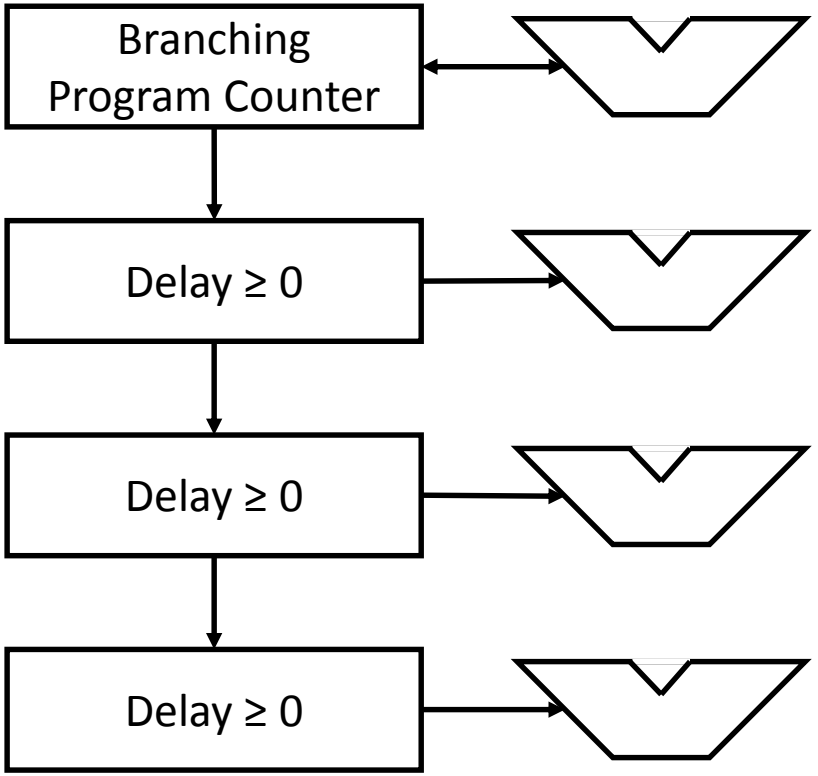
- II of 3 using 8 resources
- Modal operations waste issue slots

Cycle	ALU0	ALU1	ALU2	ALU3	ALU4	ALU5	ALU6	ALU7
0	R	^	ϕ	<?	+	<	||	>=
1	*	/	^	+	W		ϕ	ϕ
2	&	<!	ϕ	ϕ		ϕ	ϕ	ϕ

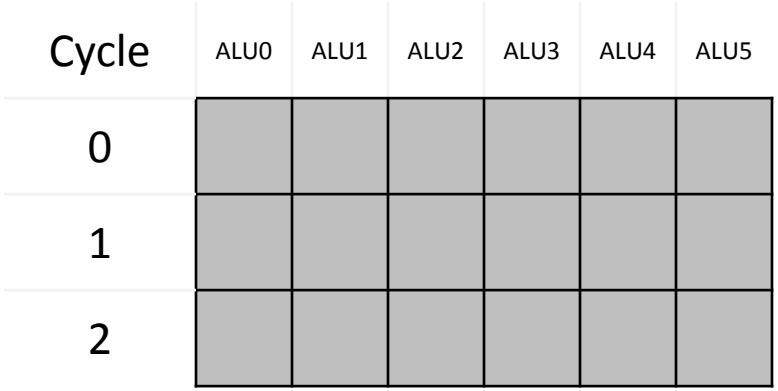
Mode 1 Operations Mode 2 Operations Control Operations

Modulo Counter Control vs Pipelined Program Counter



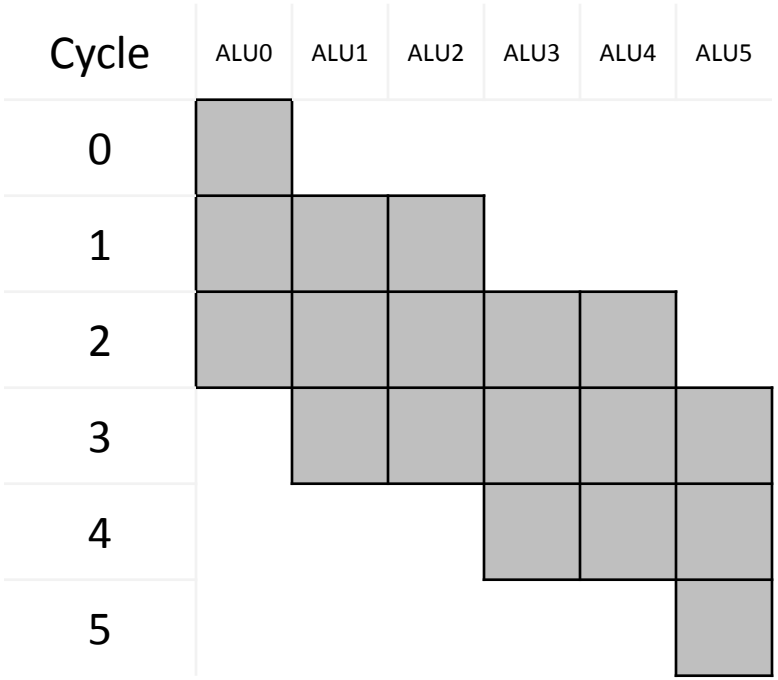


Modulo Scheduling vs Offset Pipelined Scheduling



Modulo Reservation Table

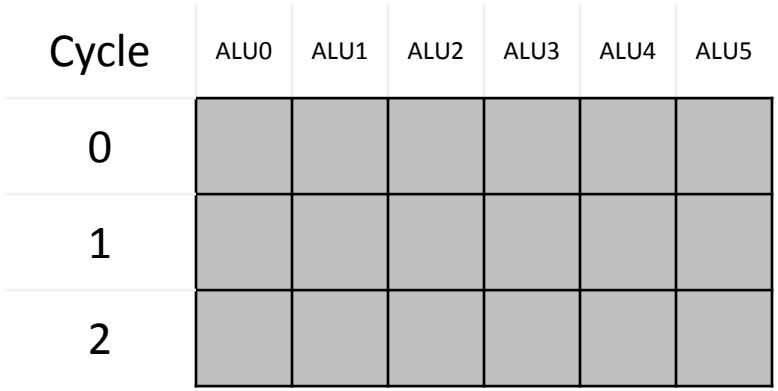
Modulo time slots

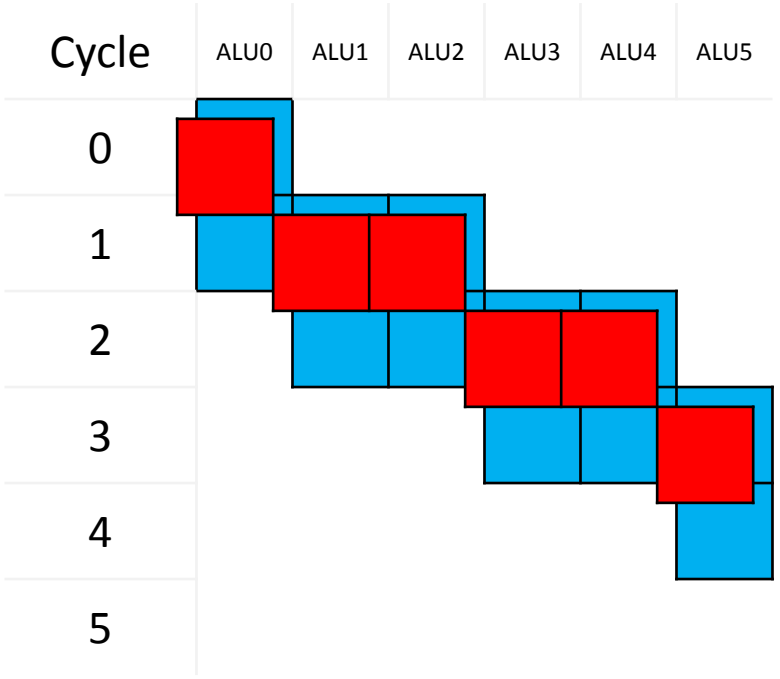


Offset Reservation Table

Fixed time slots

Modulo Scheduling vs Offset Pipelined Scheduling





Modulo Reservation Table

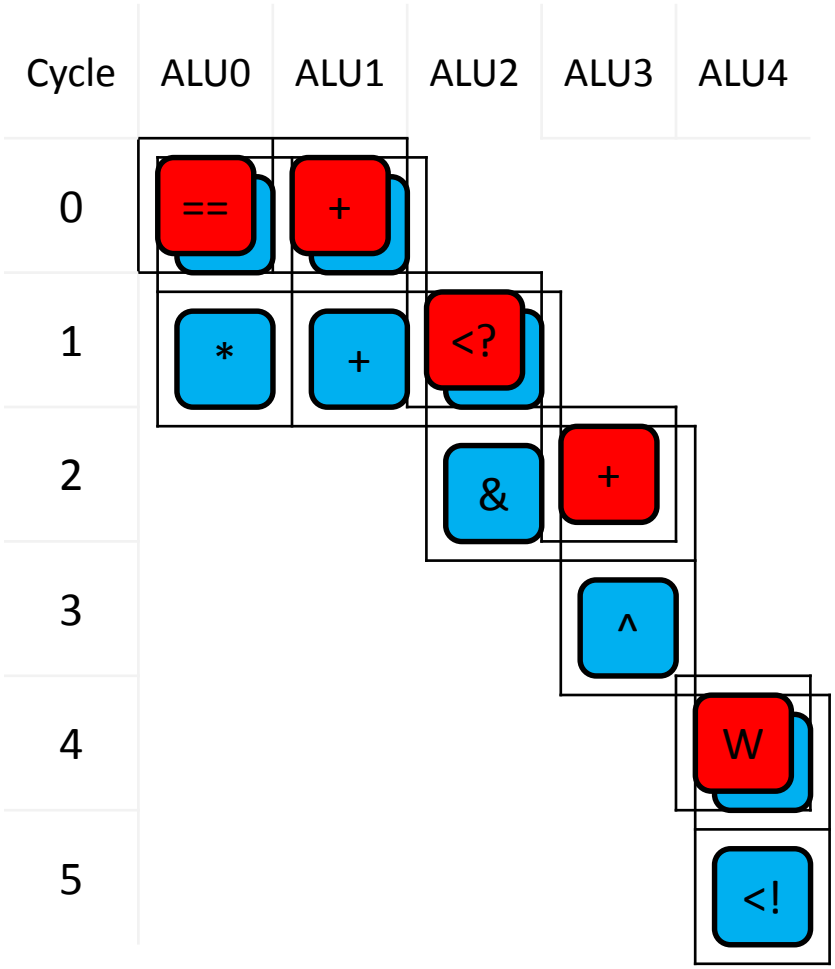
Modulo time slots

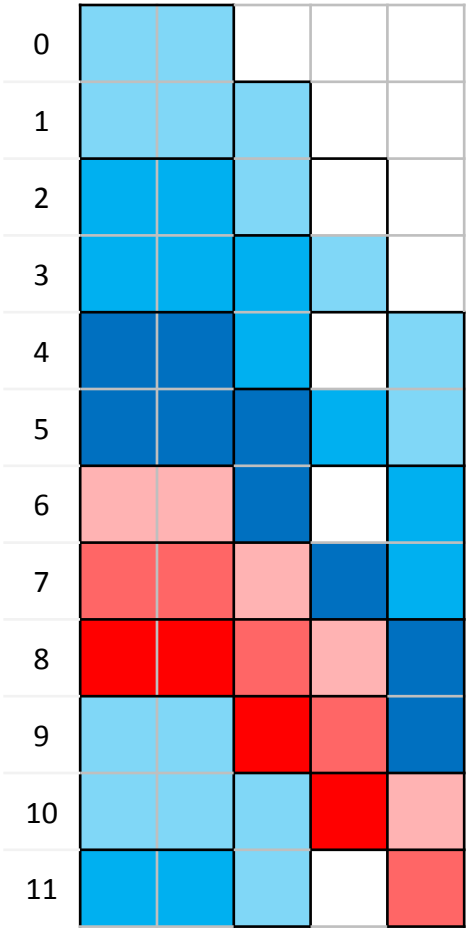
Offset Reservation Table

Fixed time slots

Offset Pipelined Schedule

- Effective II of 1.5 using 5 resources
- Reduces wasted issue slots and predicate logic





Offset Pipelined Scheduling Performance vs Iterative Modulo Scheduling Baseline