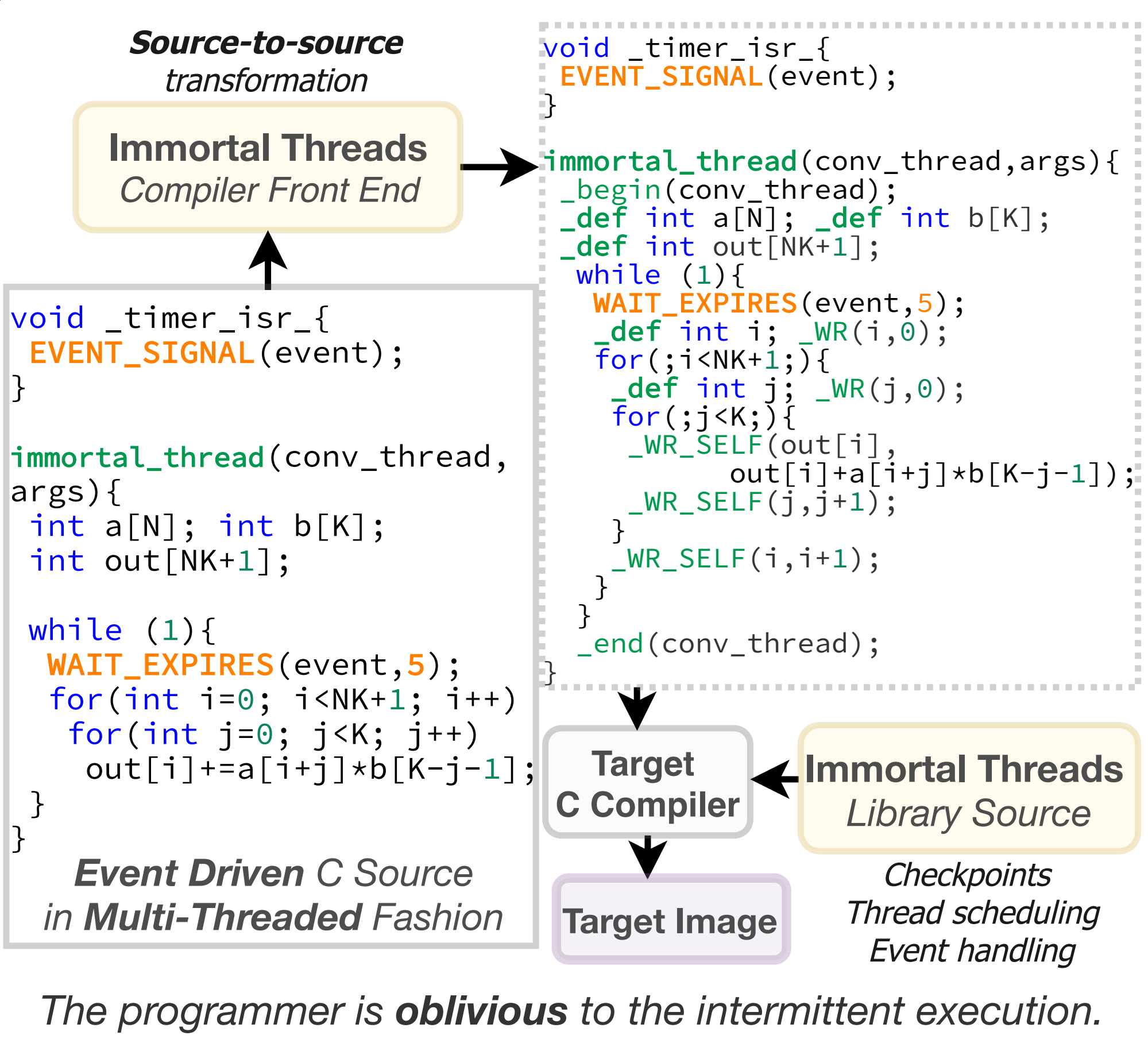
Source-to-source
transformation
Immortal Threads
Compiler Front End
void _timer_isr_{
EVENT_SIGNAL(event);
}
immortal_thread(conv_thread,args){
_begin(conv_thread);
_def int a[N]; _def int b[K];
_def int out[NK+1];
while (1){
WAIT_EXPIRES(event,5);
_def int i; _WR(i,0);
for(;i<NK+1;){
_def int j; _WR(j,0);
for(;j<K;){
_WR_SELF(out[i],
out[i]+a[i+j]*b[K-j-1]);
_WR_SELF(j,j+1);
}
_WR_SELF(i,i+1);
}
}
_end(conv_thread);
}
void _timer_isr_{
EVENT_SIGNAL(event);
}
immortal_thread(conv_thread,
args){
int a[N]; int b[K];
int out[NK+1];
while (1){
WAIT_EXPIRES(event,5);
for(int i=0; i<NK+1; i++)
for(int j=0; j<K; j++)
out[i]+=a[i+j]*b[K-j-1];
}
}
Event Driven C Source
in Multi-Threaded Fashion
Target
C Compiler
Immortal Threads
Library Source
Target Image
Checkpoints
Thread scheduling
Event handling
The programmer is oblivious to the intermittent execution.