PES UNIVERSITY

CELEBRATING 50 YEARS

Compiler Design

Preet Kanwal

Department of Computer Science & Engineering

Teaching Assistant : Kavya P K

Compiler Design

Unit 3: Static Single-Assignment(SSA)

Preet Kanwal

Department of Computer Science & Engineering

Lecture Overview

In this lecture, you will learn about -

- Static Single-Assignment (SSA) Form
- ϕ-function
- ϕ-function Examples

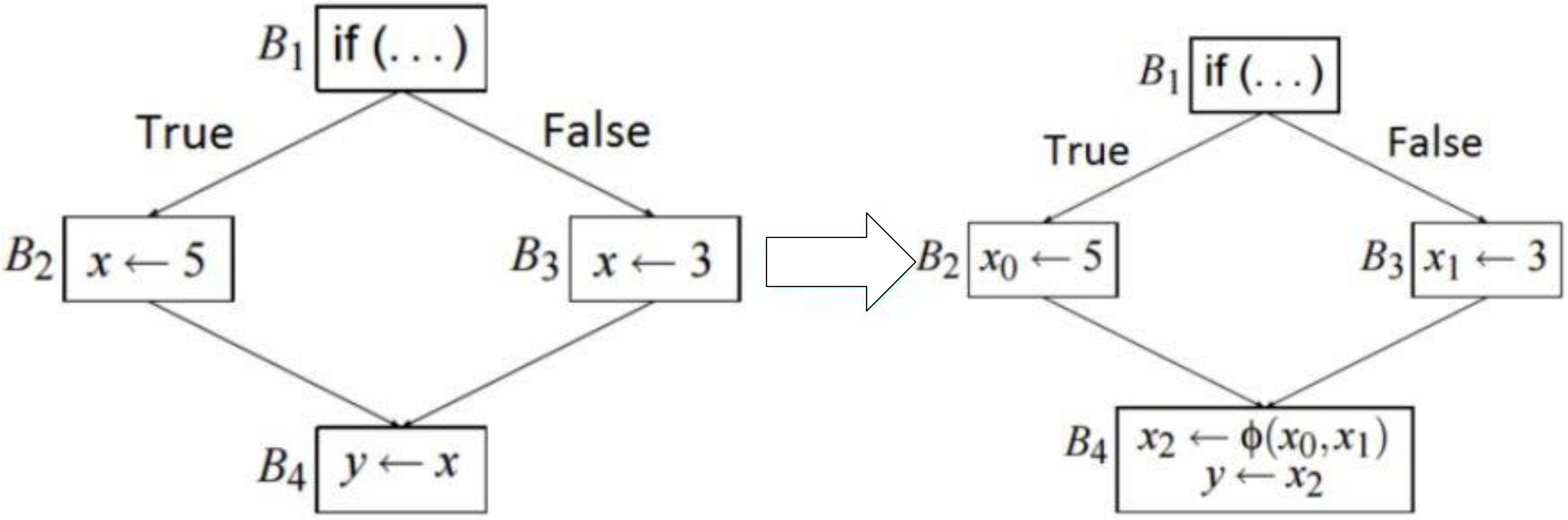
Static Single Assignment (SSA) Form

- Each variable is assigned exactly once but may be used multiple times.
- Existing variables in the original IR are split into versions:
- New version of variable is typically indicated by the original name with a subscript, so that every definition gets its own version.
- SSA is an intermediate form widely used by modern optimizing compilers.

ɸ-function

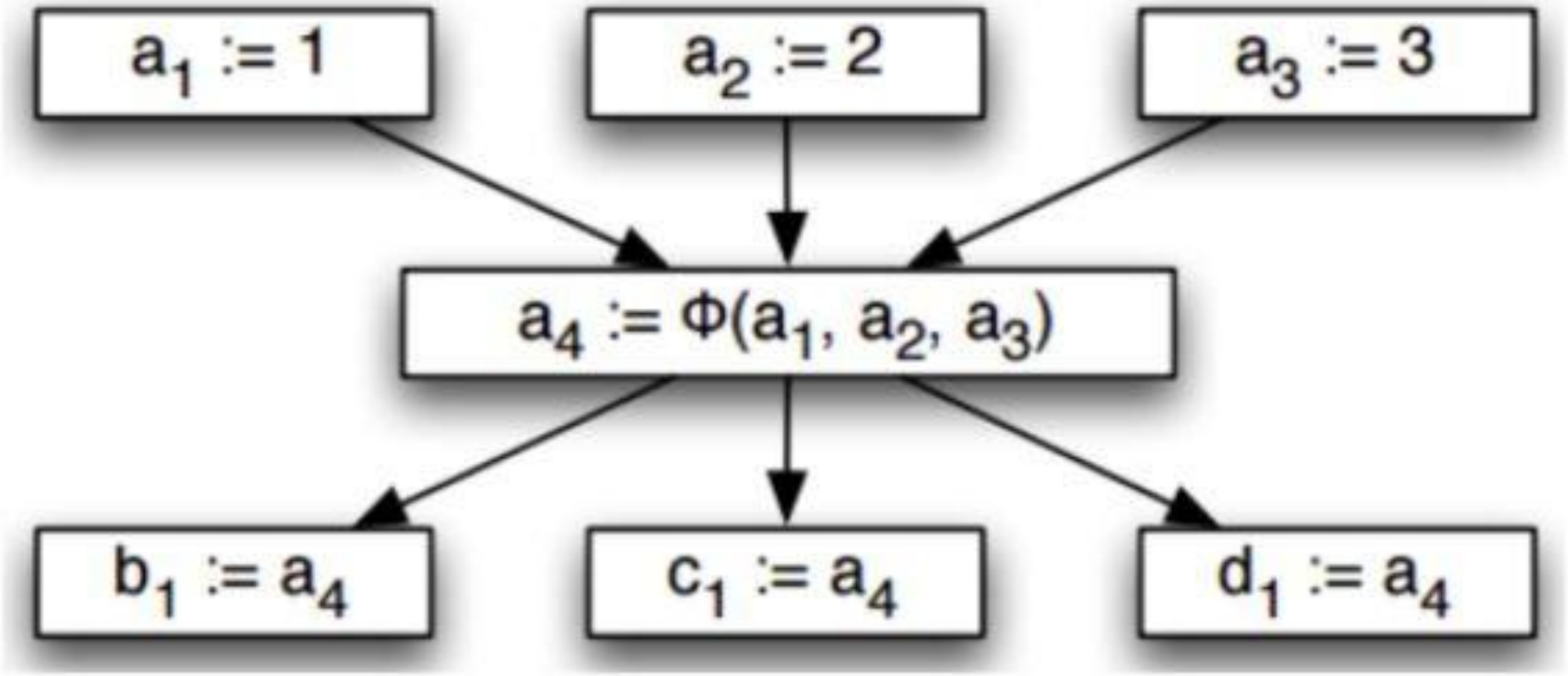
- Control flow can’t be predicted in advance, so we can’t always know which definition of a variable reached a particular use.
- To handle this uncertainty, we create ɸ functions.
- Notation represents natural “meet points” where values are combined.
- No. of arguments to ɸ(a1, a2 ....) is the number of incoming flow edges.
- Return value of the function corresponds to the control-flow path taken to get to the statement.

ϕ-function - Example 1

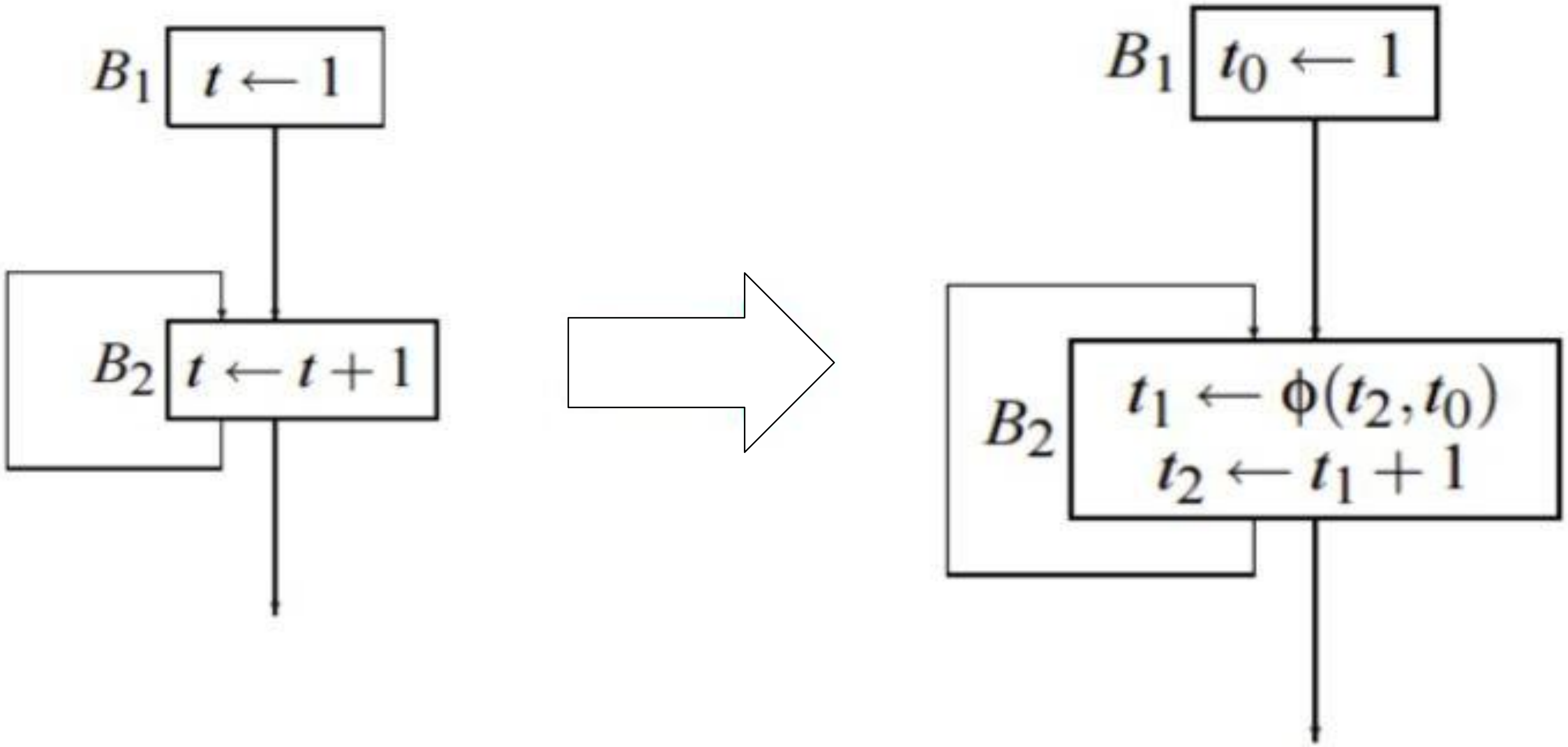


ϕ-function - Example 2

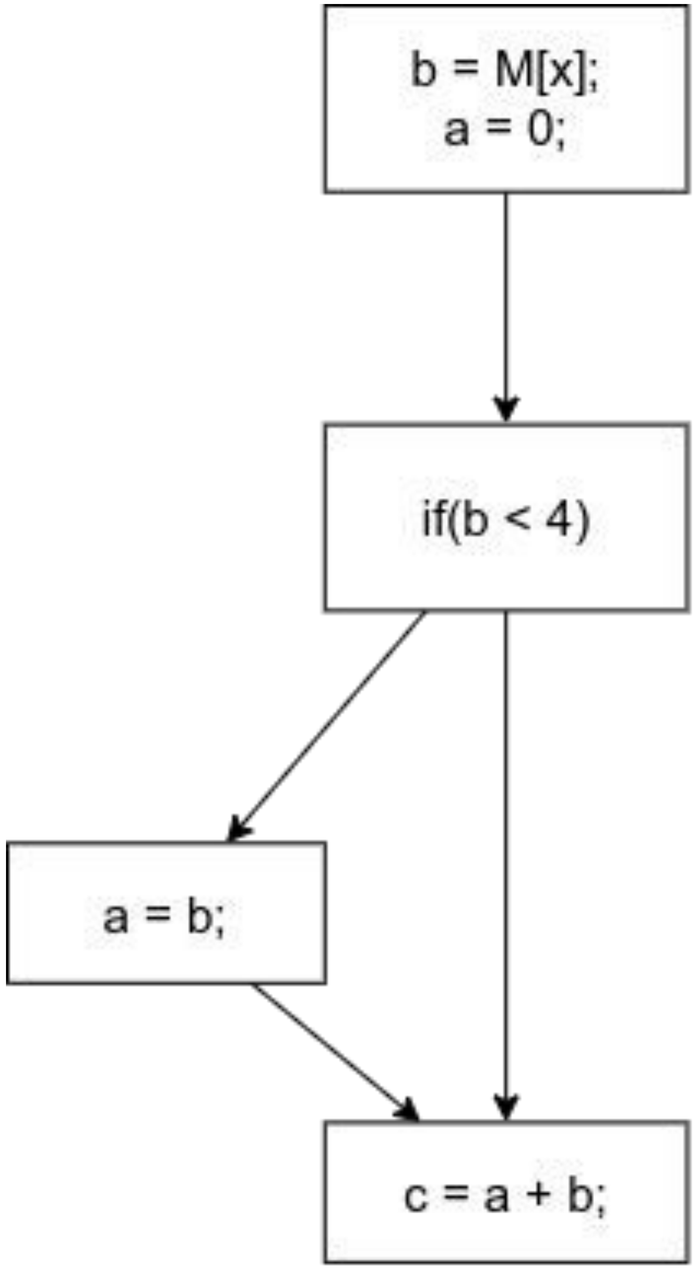
```
case (...) of
     0: a := 1;
     1: a := 2;
     2: a := 3;
end
case (...) of
     0: b := a;
     1: c := a;
     2: d := a;
end
```



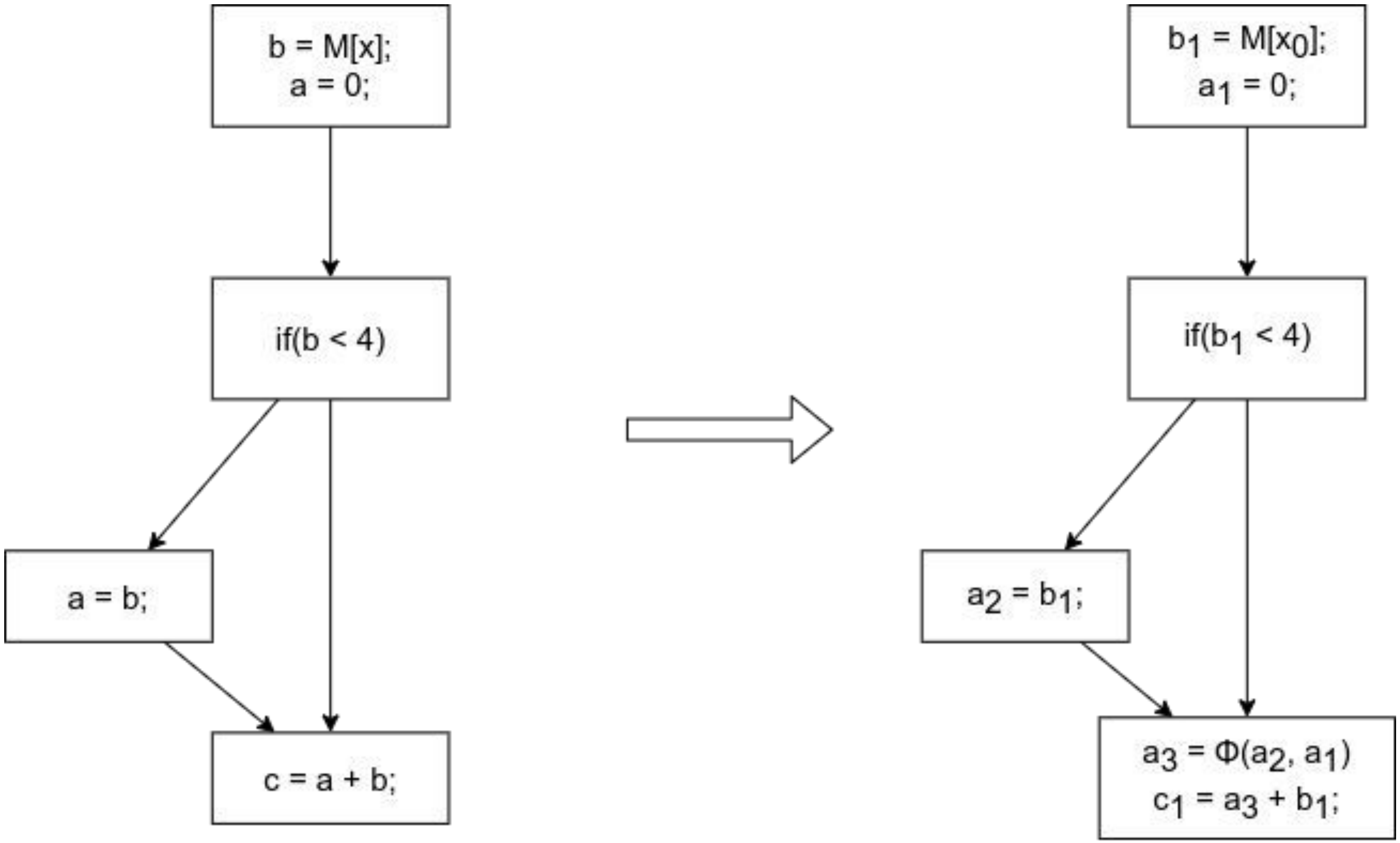
ϕ-function - Example 3



ϕ-function - Example 4



ɸ-function - Example 4



b = M[x];
a = 0;

if(b < 4)

a = b;

c = a + b;

b_1 = M[x_0];
a_1 = 0;

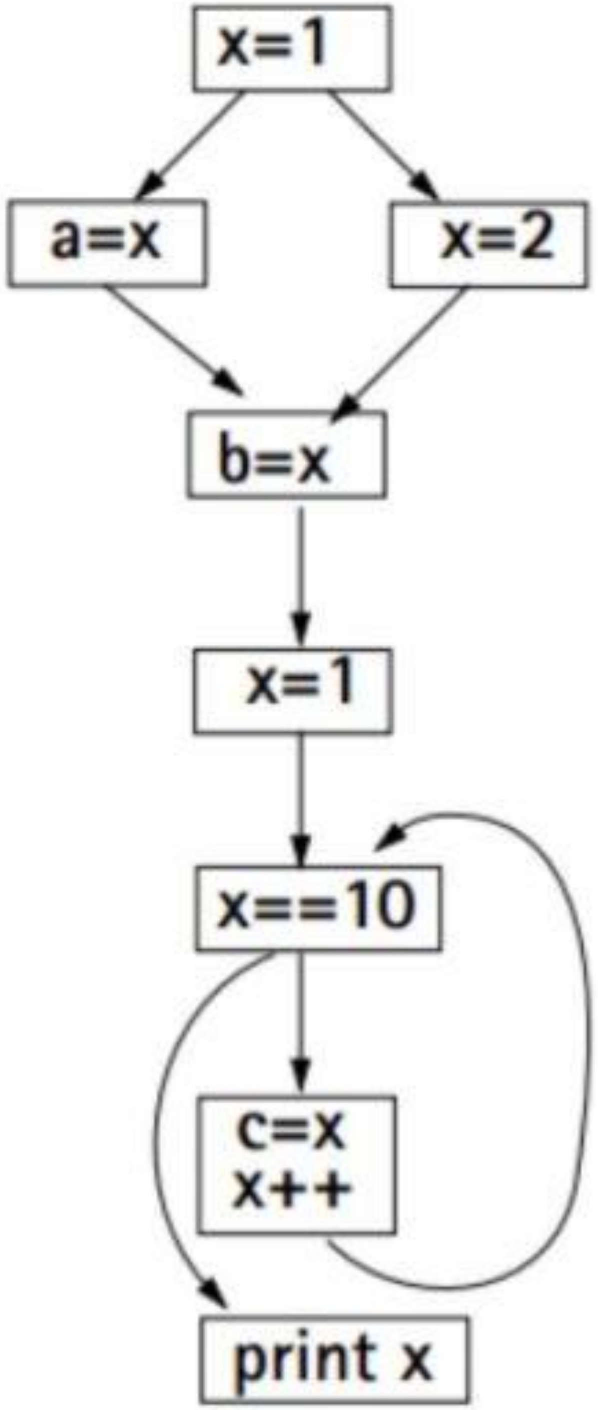
if(b_1 < 4)

a_2 = b_1;

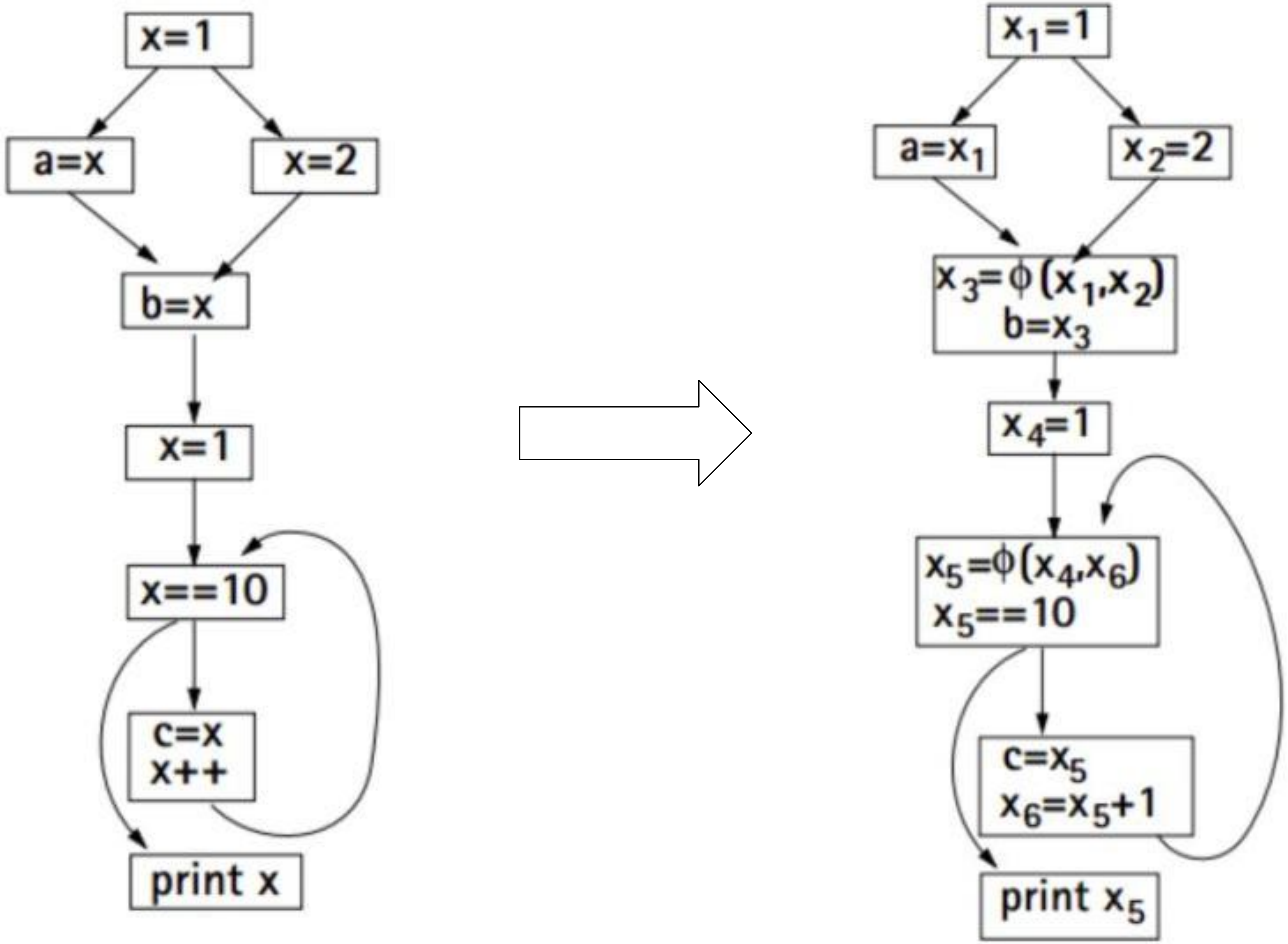
a_3 = Φ(a_2, a_1)
c_1 = a_3 + b_1;

Compiler Design

ɸ-function - Example 5



ϕ-function - Example 5



x=1

a=x

x=2

b=x

x=1

x==10

c=x
x++

print x

x_1=1

a=x_1

x_2=2

x_3=ϕ(x_1,x_2)
b=x_3

x_4=1

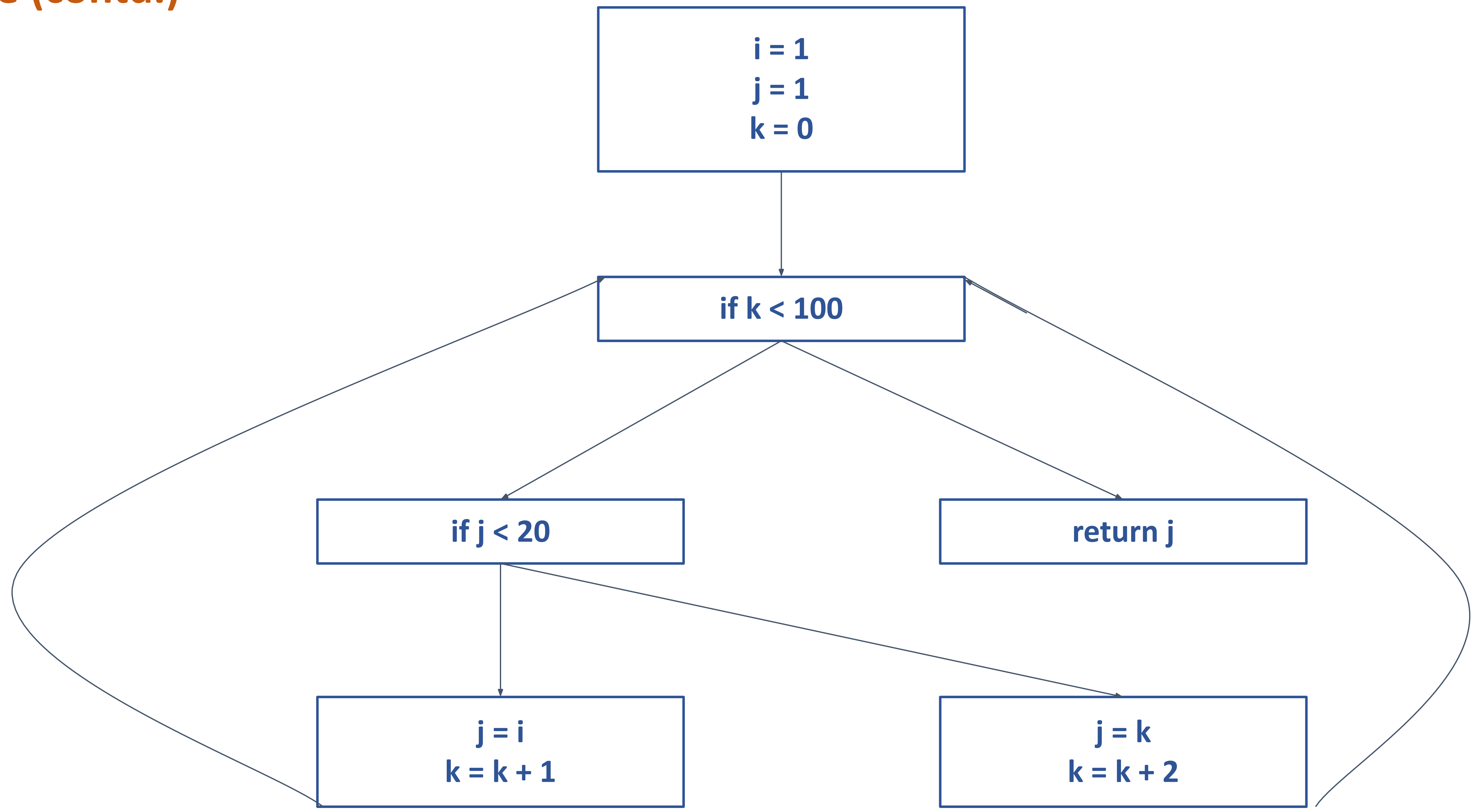
x_5=ϕ(x_4,x_6)
x_5==10

c=x_5
x_6=x_5+1

print x_5

ϕ-function - Example 6

Example (contd.)

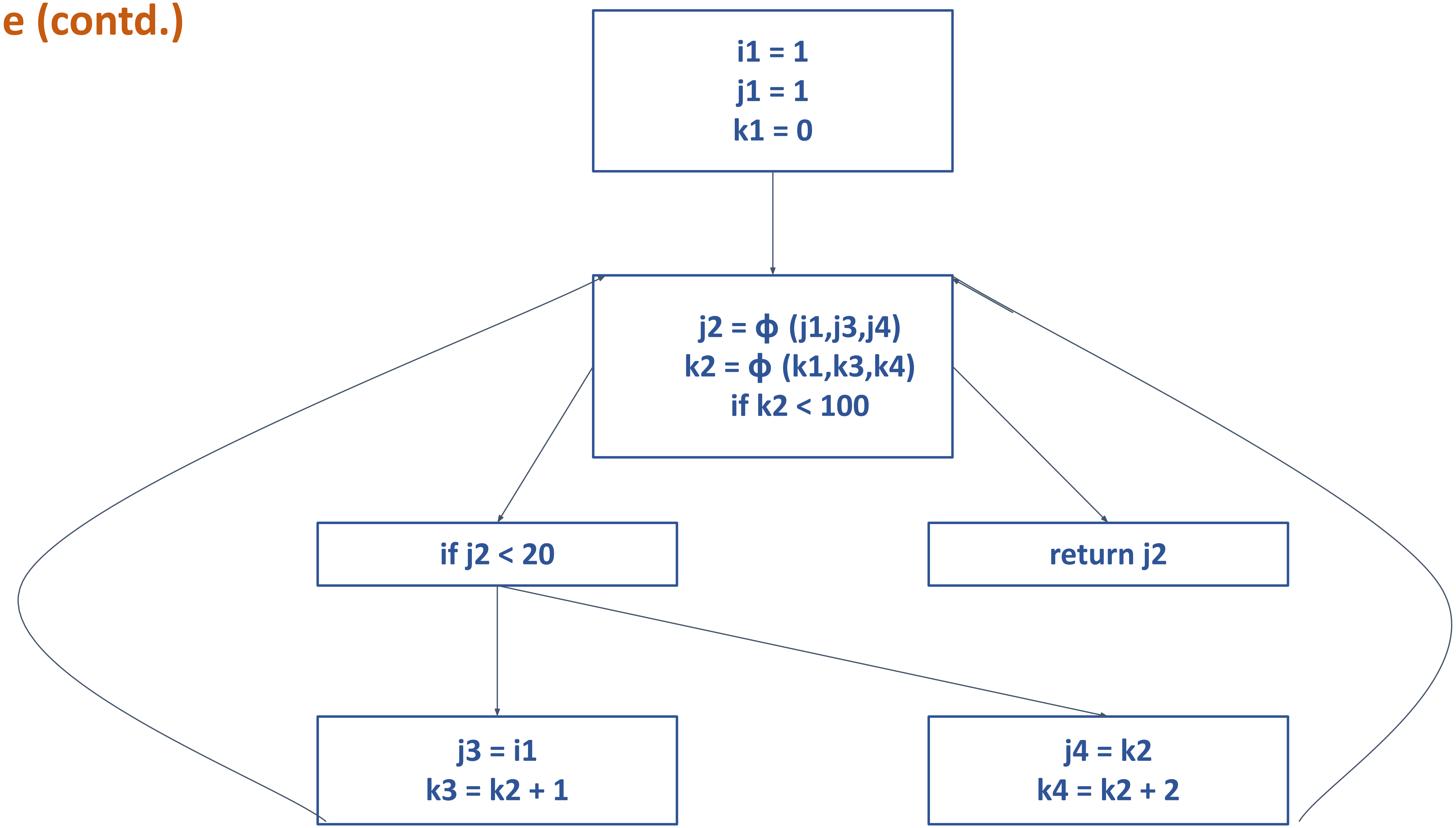


Compiler Design

ϕ-function - Example 6

Example (contd.)

SSA



i1 = 1
j1 = 1
k1 = 0

j2 = ϕ (j1,j3,j4)
k2 = ϕ (k1,k3,k4)
if k2 < 100

if j2 < 20

return j2

j3 = i1
k3 = k2 + 1

j4 = k2
k4 = k2 + 2

ɸ-function - Example 7

Example

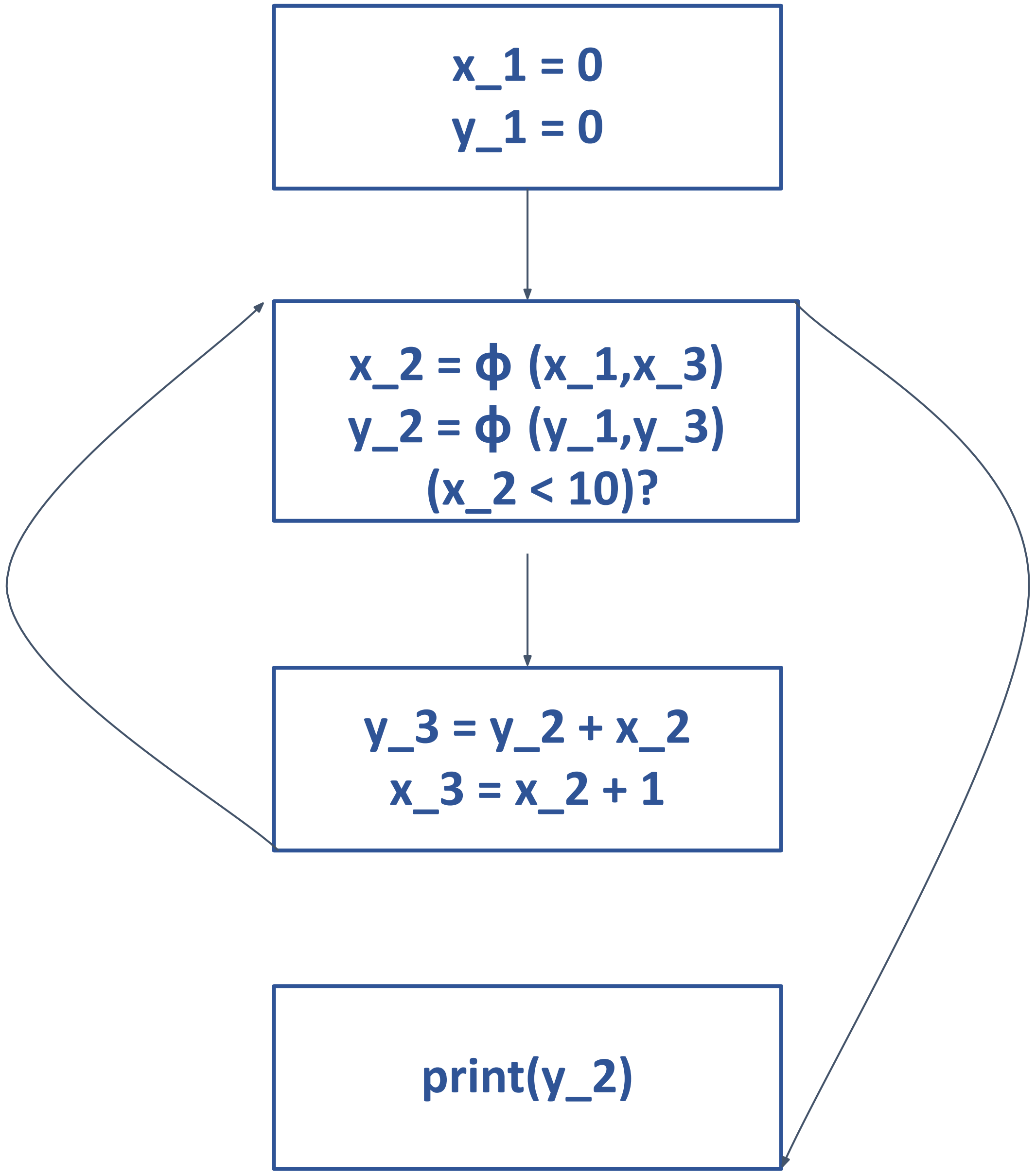
```
x = 0;
y = 0;
while (x<10){
y = y+x;
x = x+1;
}
print(y);
```

Compiler Design

ϕ-function - Example 7

Example

```
x = 0;

y = 0;
while (x<10){

y = y+x;

x = x+1;

}

print(y);
```



Note -

- Most modern production compilers use SSA form (eg. gcc, suif, llvm, hotspot etc.)
- Popular compiler optimizations (eg. constant propagation) become easier to write (and in some cases, algorithmically faster) when applied to programs in SSA form.
- Conversion to SSA form introduces a lot of assignments - compilers that do this need to have good register allocators that can eliminate most of them again (not a concern these days).

PES
UNIVERSITY
CELEBRATING 50 YEARS

THANK YOU

Preet Kanwal

Department of Computer Science & Engineering

preetkanwal@pes.edu