# Loop Invariant Computations

M.B.Chandak

# Background

- Convert TAC to PFG
- Compute Reaching definitions
- Compute u-d chain
- Compute Dominators
- Compute Back edges and detect loops
- Compute loop invariants using u-d chain information

- *How to move loop invariant out of loop ➔ → Process in next slides*

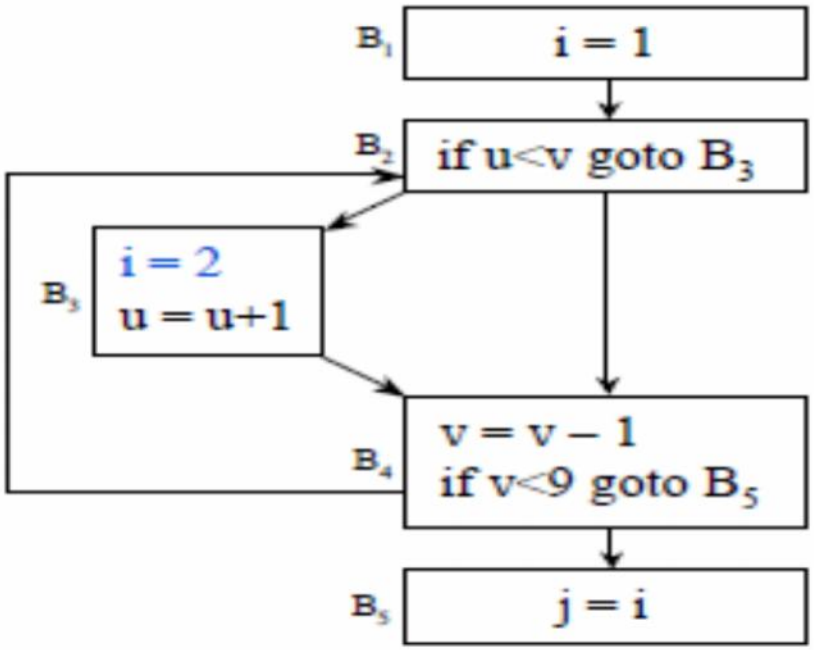
# Steps: For code motion in loop invariant.

1. Find loop invariant statements
2. For each statement “s” defining “x” check:
    1. It is in the block that dominates all exists of “Loop”
    2. “X” is not defined elsewhere in “L”
    3. All uses of “x” in loop can only reach by definition of “x” in “s”.
3. Move each statement of “s” found in step (1) and satisfying all conditions of step (2) to a newly created pre-header block.
4. Update the u-d chain accordingly.

# Steps: Condition 2.1

**Condition 1 is Needed**

- If the block containing s does not dominate all exits, after the loop might get a different value



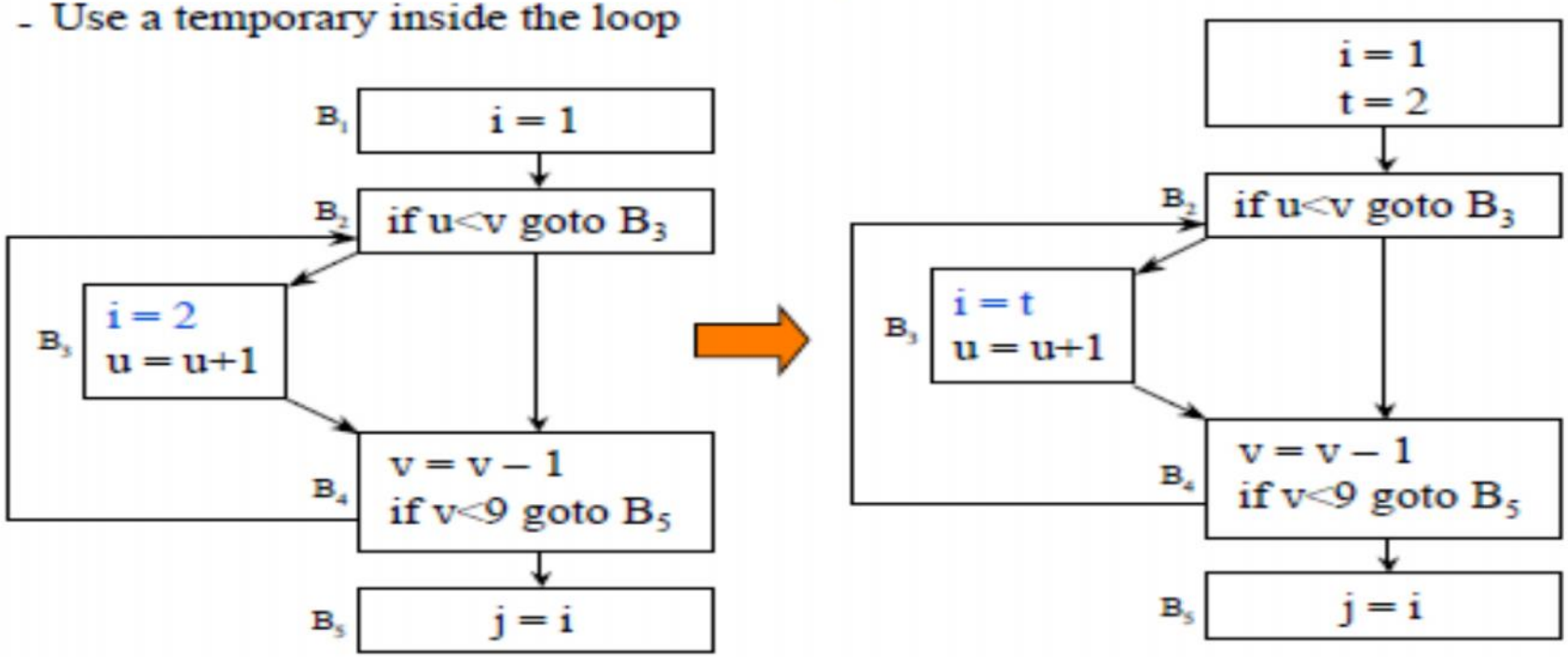

Can we move i=2 outside the loop?

i=2 is loop invariant, but $B_3$ does not dominate $B_4$, the exit node, so moving i=2 would change the meaning of the loop for those cases where $B_3$ is never executed

# Condition: 2.1 [Modified]

**Using the alternate approach**

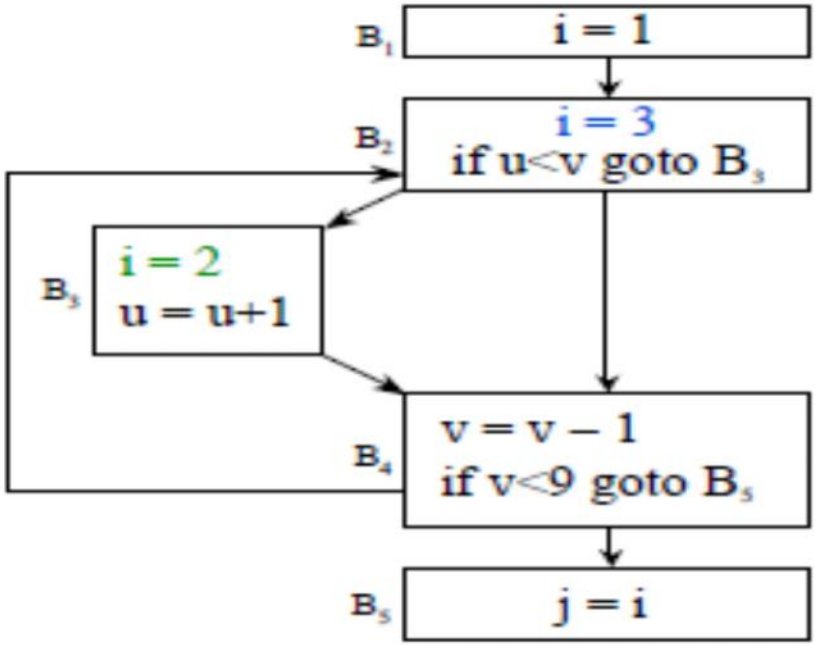
- Move the invariant code outside the loop
- Use a temporary inside the loop



# Steps: Condition: 2.2

**Condition 2 is Needed**

- If some other statement in the loop assigns i, the movement of the statement may cause some statement to see the wrong value



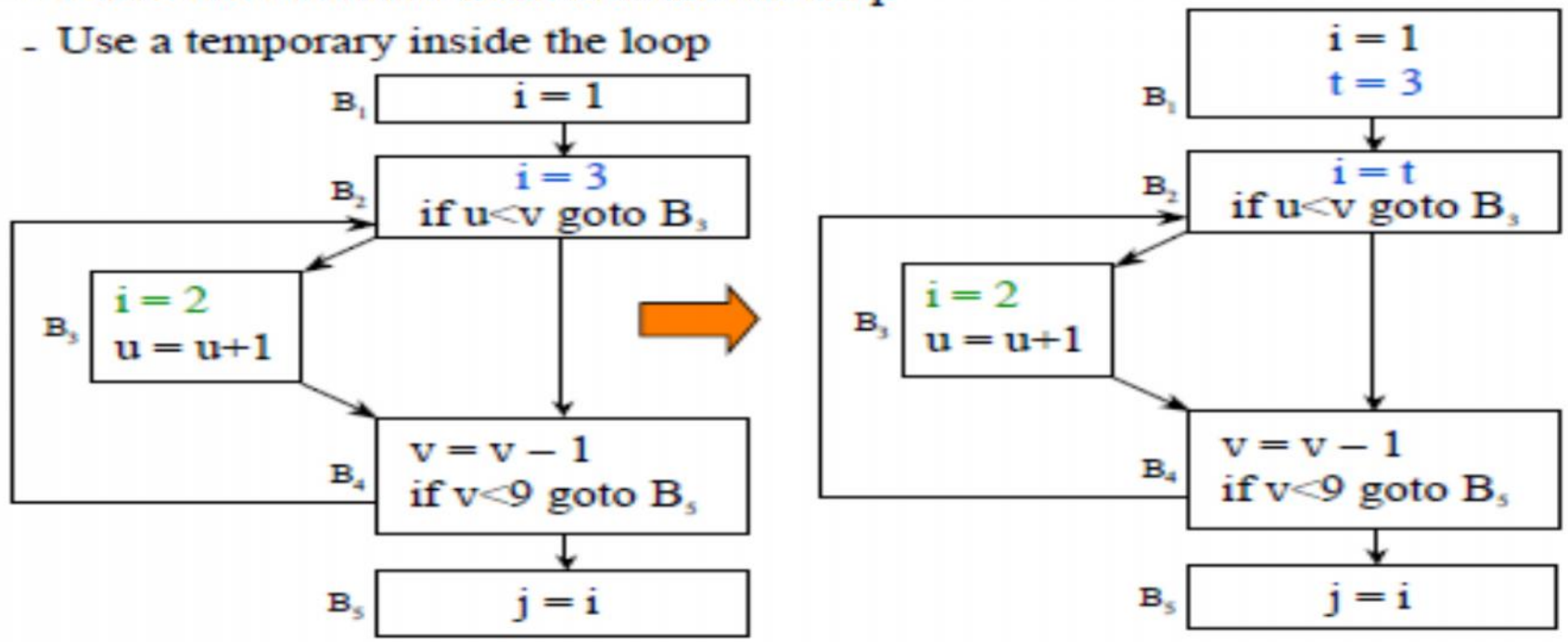

Can we move i=3 outside the loop?

$B_2$ dominates the exit so condition 1 is satisfied, but code motion will set the value of i to 2 if $B_3$ is ever executed, rather than letting it vary between 2 and 3.

# Condition 2.2: Modified

**Using the alternate approach**

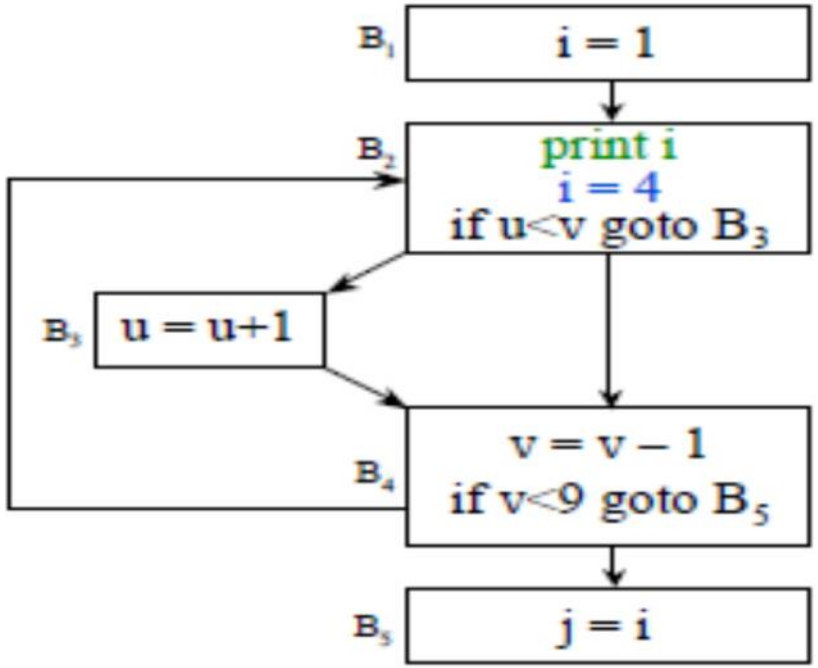
- Move the invariant code outside the loop
- Use a temporary inside the loop



# Steps: Condition 2.3

**Condition 3 is Needed**

- If a use in L can be reached by some other def, then we cannot move the def outside the loop




Can we move i=4 outside the loop?

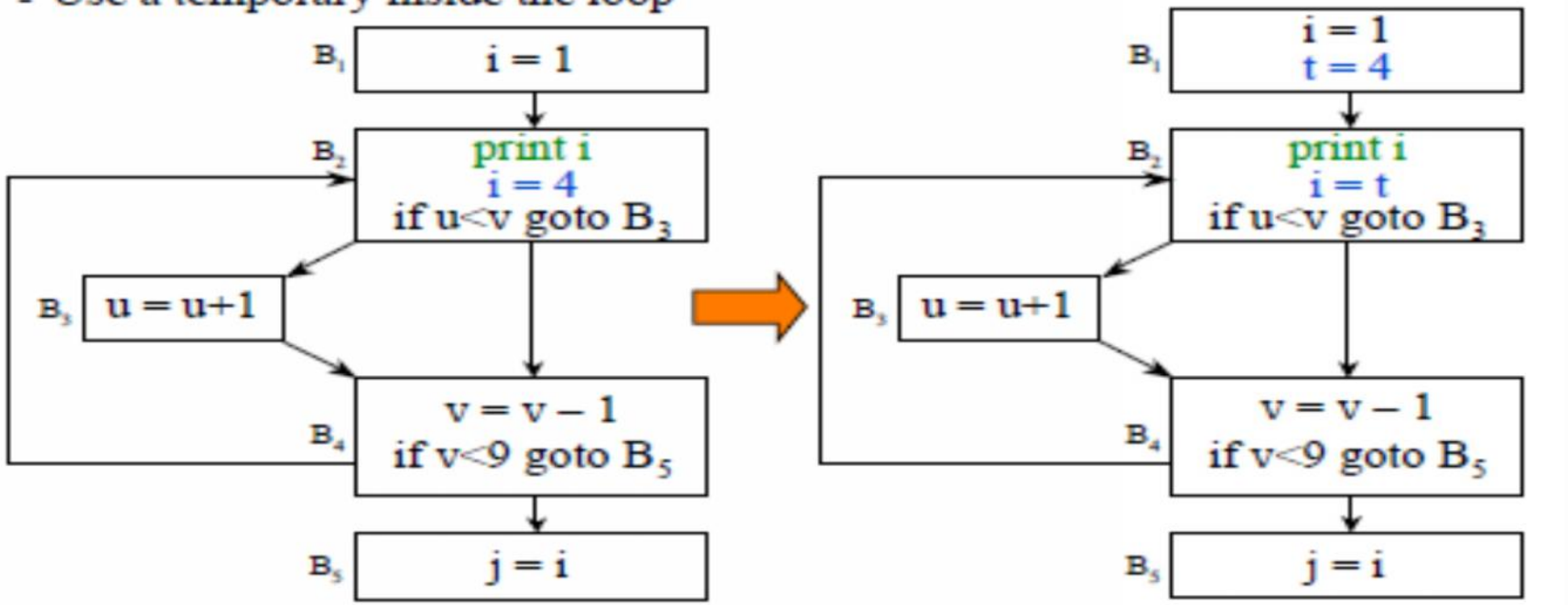
Conditions 1 and 2 are met, but the use of i in block $B_2$, can be reached from a different def, namely i=1 from $B_1$.

If we were to move i=4 outside the loop, the first iteration through the loop would print 4 instead of 1

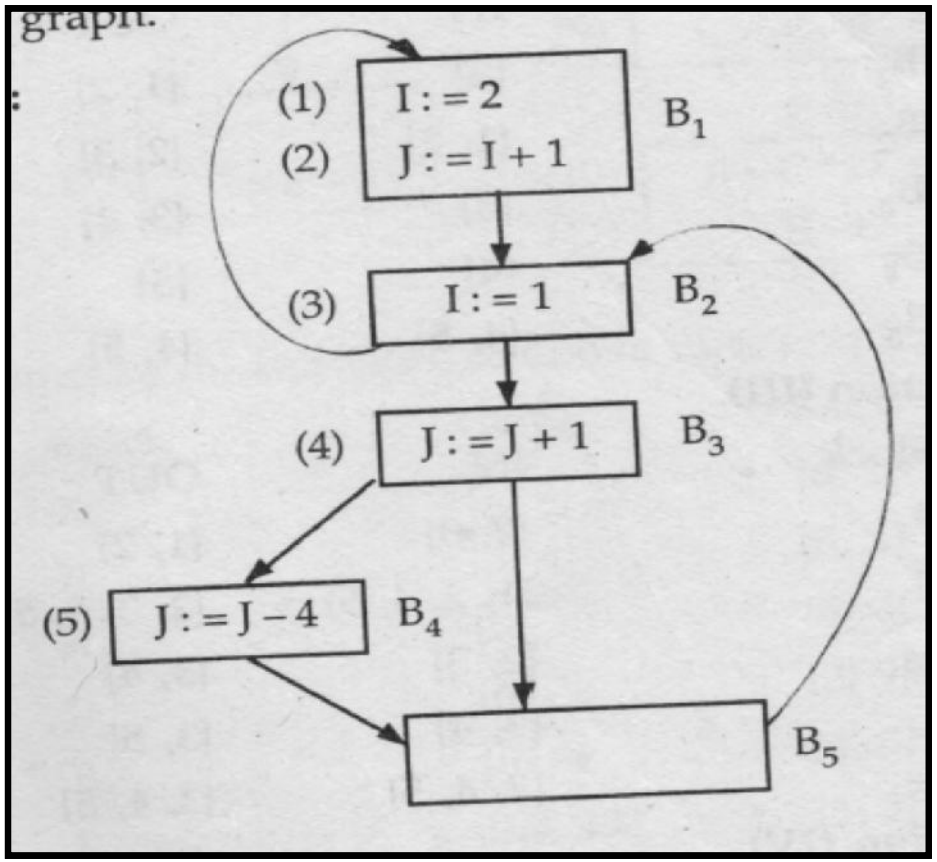
# Condition 2.3: Modified

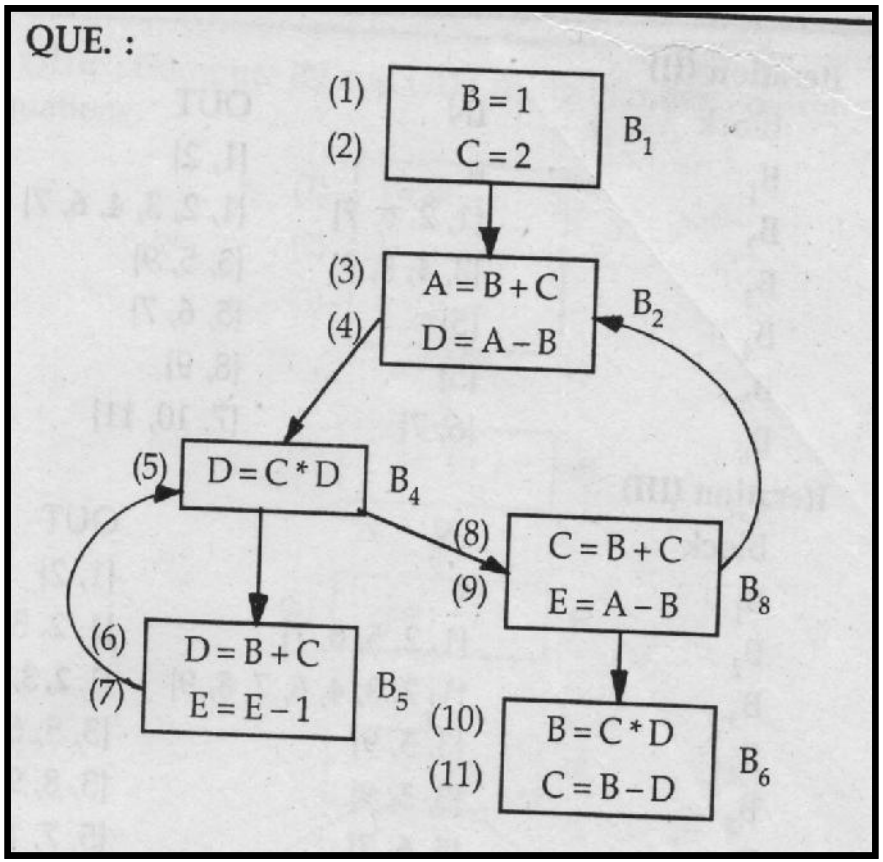
**Using the alternate approach**

- Move the invariant code outside the loop
- Use a temporary inside the loop



# Assignment Questions: Reaching Definition, Live Variable Loop invariant computation



# Next Elimination of Induction Variable

M.B.Chandak