



# Gcc Loop Representation

*Dominique Heller*

*Dominique.heller@univ-ubs.fr*

**CNRS, UMR 3192, Lab-STICC  
Centre de Recherche - BP 92116  
F-56321 Lorient Cedex - FRANCE**

---

# Plan

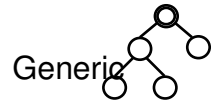
- Gcc Loop representation
  - For/while
  - Do while
  - No natural form (break, continue)

# Loop representation : for/do while

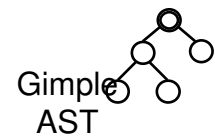
```
int main(int a,int *s)
{
  int i; int sum = 0;
  for (i = 0; i < 1000; i++)
    sum += a ;
  *s=sum;
  return 0;
}
```

```
int main(int a,int *s)
{
  int i = 0; int sum = 0;
  do {
    sum += a ;
    i++;
  } while (i < 1000);
  *s=sum;
  return 0;
}
```

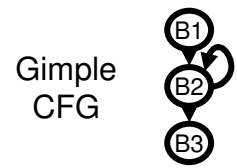
Front End



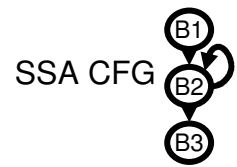
Gimplifier



CFG builder



IPA SSA



Loop Reconstruction

-fdump-tree-loopinit

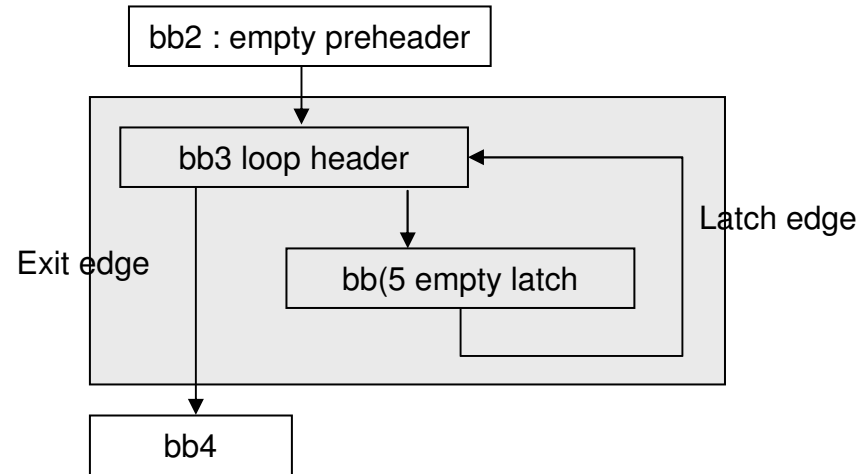
```
;; Function int main(int, int*) (main)
;; Loop 1
;; header 3, latch 5
;; nodes: 3 5
int main(int, int*) (a, s)
{
  int sum;int i;
<bb 2>:
<bb 3>:
  # sum_17 = PHI <sum_6(5), 0(2)>
  # i_16 = PHI <i_7(5), 0(2)>
  sum_6 = sum_17 + a_5(D);
  i_7 = i_16 + 1;
  if (i_7 <= 999)
    goto <bb 5>;
  else
    goto <bb 4>;
<bb 5>:
  goto <bb 3>;
<bb 4>:
  # sum_13 = PHI <sum_6(3)>
  *s_8(D) = sum_13;
  return 0;
}
```



# Loop representation : for/do while

-fdump-tree-loopinit

```
;; Function int main(int, int*) (main)
;; Loop 1
;; header 3, latch 5
;; nodes: 3 5
int main(int, int*) (a, s)
{
  int sum;int i;
  <bb 2>:
  <bb 3>:
    # sum_17 = PHI <sum_6(5), 0(2)>
    # i_16 = PHI <i_7(5), 0(2)>
    sum_6 = sum_17 + a_5(D);
    i_7 = i_16 + 1;
    if (i_7 <= 999)
      goto <bb 5>;
    else
      goto <bb 4>;
  <bb 5>:
    goto <bb 3>;
  <bb 4>:
    # sum_13 = PHI <sum_6(3)>
    *s_8(D) = sum_13;
    return 0;
}
```



Single latch

Single exit edge : natural form

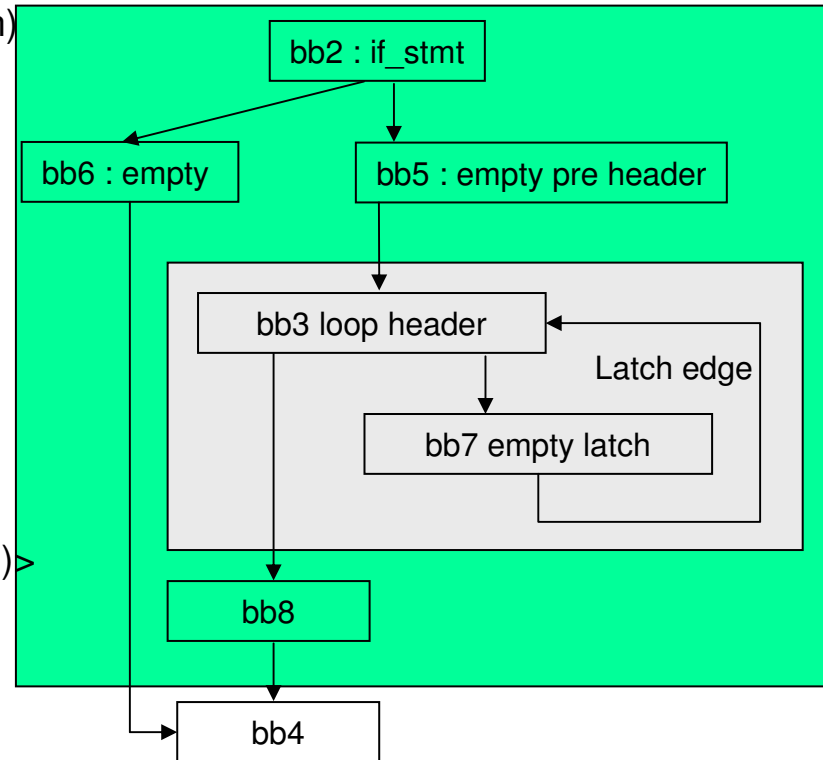
Induction variable i, chrec : {0,+,-1}

Cond Expr (i<=999) in loop header

# Loop representation : while = if + do while

```
int main(int a,int *s)
{
  int i ;
  int sum = 0;
  while (i < 1000)
  {
    sum += a ;
    i++;
  }
  *s=sum;
  return 0;
}
```

```
;; Function int main(int, int*) (main)
;; Loop 1 header 3, latch 7
;; nodes: 3 7
int main(int, int*) (a, s)
{
  int sum;int i;
<bb 2>:
  if (i_4(D) <= 999) goto <bb 5>;
  else goto <bb 6>;
<bb 6>: goto <bb 4>;
<bb 5>:
<bb 3>:
  # sum_17 = PHI <sum_6(7), 0(5)>
  # i_16 = PHI <i_7(7), i_4(D)(5)>
  sum_6 = sum_17 + a_5(D);
  i_7 = i_16 + 1;
  if (i_7 <= 999) goto <bb 7>;
  else goto <bb 8>;
<bb 7>: goto <bb 3>;
<bb 8>:
  # sum_1 = PHI <sum_6(3)>
<bb 4>:
  # sum_13 = PHI <sum_1(8), 0(6)>
  *s_8(D) = sum_13;
  return 0;
}
```



Single latch

Single exit edge : natural form

Induction variable i, chrec : {0,+ ,1}

Cond Expr (i<=999) in loop header

## Loop representation : break, no natural form ?

```
int main(int a,int *s)
{
  int i;
  int sum = 0;
  for (i = 0; i < 1000; i++)
  {
    sum += a ;
    if (sum>100)
      break;
  }
  *s=sum;
  return 0;
}
```

```
:: Function int main(int, int*) (main)
```

```
:: Loop 1
```

```
:: header 4, latch 9
```

```
:: nodes: 4 9 3
```

```
int main(int, int*) (a, s)
```

```
{
```

```
  int sum;int i;
```

```
<bb 2>:
```

```
  if (a_6(D) > 100) goto <bb 6>;
  else goto <bb 7>;
```

```
<bb 7>: goto <bb 4>;
```

```
<bb 6>: goto <bb 5>;
```

```
<bb 3>:
```

```
  sum_7 = sum_18 + a_6(D);
  if (sum_7 > 100) goto <bb 8>;
  else goto <bb 9>;
```

```
<bb 8>:
```

```
  # sum_1 = PHI <sum_7(3)>
  goto <bb 5>;
```

```
<bb 9>:
```

```
<bb 4>:
```

```
  # sum_18 = PHI <sum_7(9),
  a_6(D)(7)>
  # i_17 = PHI <i_8(9), 0(7)>
  i_8 = i_17 + 1;
  if (i_8 <= 999) goto <bb 3>;
  else
    goto <bb 10>;
```

```
<bb 10>:
```

```
  # sum_16 = PHI <sum_18(4)>
```

```
<bb 5>:
```

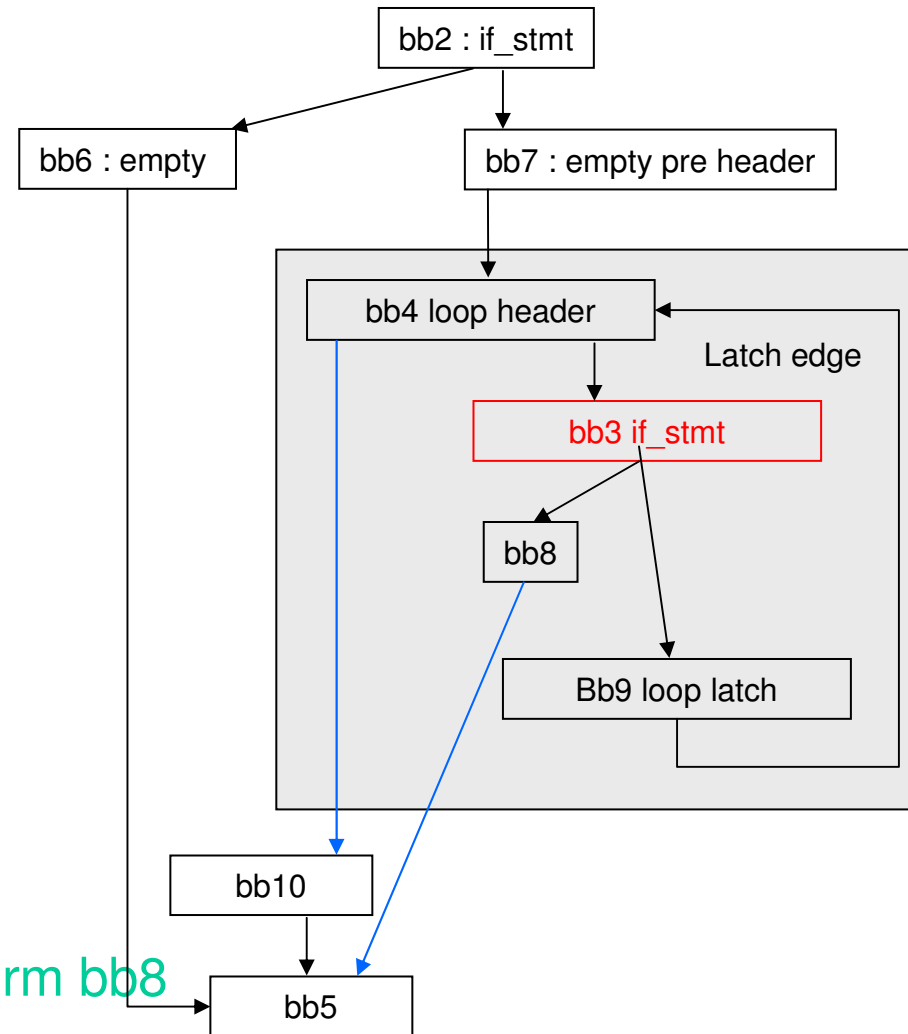
```
  # sum_3 = PHI <sum_1(8),
  sum_16(10), a_6(D)(6)>
  *s_9(D) = sum_3;
  return 0;
```

```
  return 0;
```

```
}
```

## Loop representation : break, no natural form ?

```
int main(int a,int *s)
{
  int i;
  int sum = 0;
  for (i = 0; i < 1000; i++)
  {
    sum += a ;
    if (sum>100)
      break;
  }
  *s=sum;
  return 0;
}
```



Single latch

two exit edge

No natural form : bb4 not reachable form bb8

How to represent break/continue in a hierarchical cdfg ?