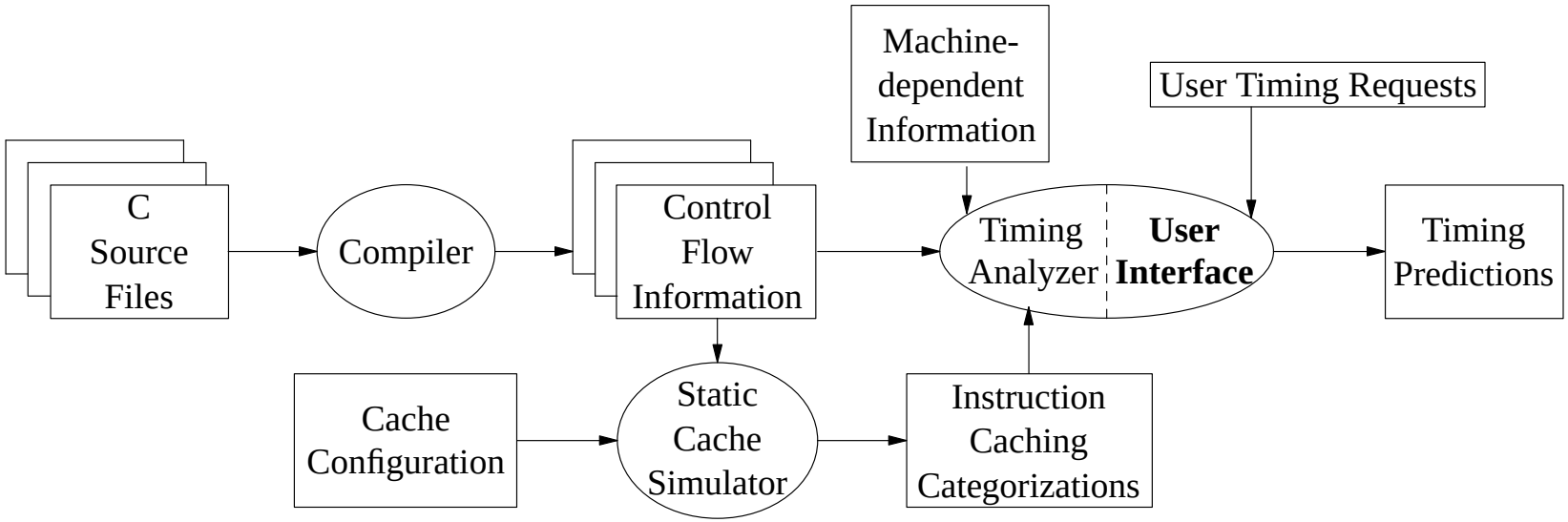
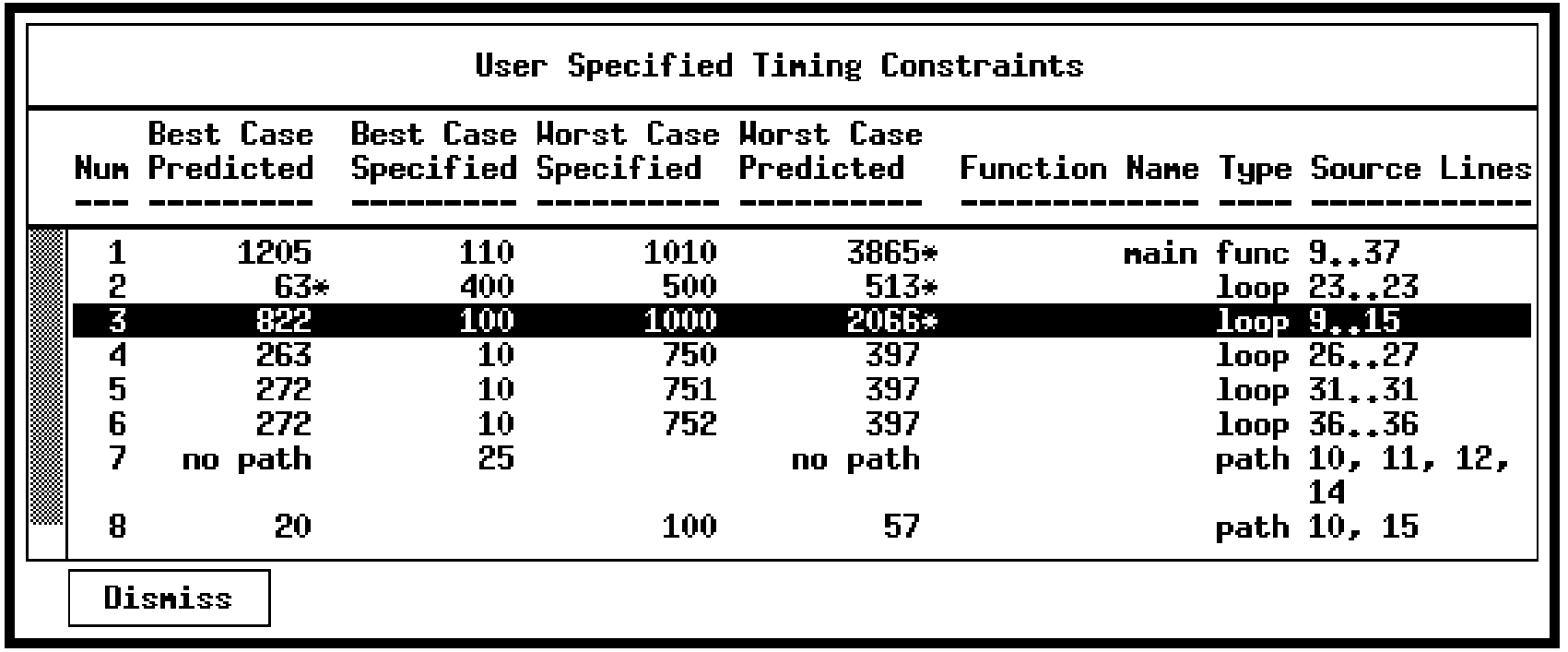
Overview of Obtaining Timing Predictions



Source Code with Timing Constraints

```
 1 functimebnd [:2ms]
 2 void Sum(Array, Nonnegcnt, Negcnt, nonnegsum, Negsum)
 3 matrix Array;
 4 int *Nonnegcnt, *Negcnt, *Nonnegsum, *Negsum;
 5 {
 6    int i, j;
 7    void Addnonneg(), Addneg();
 8
 9    *Nonnegsum = *Negsum = *Nonnegcnt = *Negcnt = 0;
10    for (i=1; i <= MAXSIZE; i++)
11       looptimebnd [500ns:3us]
12       for (j=1; j <= MAXSIZE; j++)
13          if (Array[i][j] >= 0) {     @1[:150ns] @2[10ns:100ns]
14             Addnonneg(Array[i][j], Nonnegsum);
15             (*Nonnegcnt)++;          @1
16             }
17          else {
18             *Negsum += Array[i][j];  @2
19             (*Negcnt)++;             @2
20             }
21 }
```

Constraints Window



User Specified Timing Constraints

Num	Best Case Predicted	Best Case Specified	Worst Case Specified	Worst Case Predicted	Function Name	Type	Source Lines
1	1205	110	1010	3865*	main	func	9..37
2	63*	400	500	513*		loop	23..23
3	822	100	1000	2066*		loop	9..15
4	263	10	750	397		loop	26..27
5	272	10	751	397		loop	31..31
6	272	10	752	397		loop	36..36
7	no path	25		no path		path	10, 11, 12, 14
8	20		100	57		path	10, 15

Dismiss