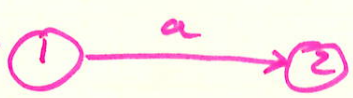
CONTROL EDGES & DATA EDGES

A CONTROL EDGE IS A RELATION BETWEEN TWO OPERATIONS SUCH THAT ONE OP IMMEDIATELY FOLLOWS THE OTHER

A DATA EDGE IS A RELATION BETWEEN TWO OPERATIONS SUCH THAT DATA PRODUCED BY ONE OP IS CONSUMED BY THE OTHER

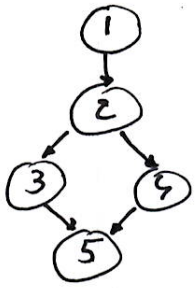



```
a = 15      // 1
b = a + 2;  // 2
```

Control Edges

```
int max(int a, int b) {
  if (a > b)
    r = a;
  else
    r = b;
  return r;
}
```

NODES	TO	TO
1	2	
2	3	4
3	5	
4	5	
5	EXIT	



Data R/W

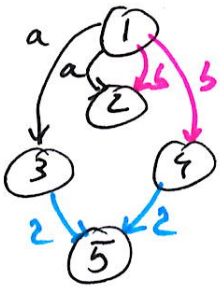
```
int max(int a, int b) {
  if (a > b)
    r = a;
  else
    r = b;
  return r;
}
```

	a	b	c
1	W	W	
2	R	R	
3	R		W
4		R	W
5			R

Data Edges

```
int max(int a, int b) {
  if (a > b)
    r = a;
  else
    r = b;
  return r;
}
```

	a	b	r2
1			
2			
3			
4			
5			



Control Edges/ Data R/W

```
max(r4, r5):
  mov r2, r4
  bge r4, r5, .L2
  mov r2, r5
.L2:
  ret (r2)
```

	TO	R4	R5	R2
1	2	W	W	
2	3	R		W
3	4, 5	R	R	
4	5		R	W
5	EXIT			R

Data Edges

```
max(r4, r5):
  mov r2, r4
  bge r4, r5, .L2
  mov r2, r5
.L2:
  ret (r2)
```

	R4	R5	R2
1			
2			
3			
4			
5			

Control Edges

```
unsigned gcd(unsigned a,
             unsigned b){
  while (a != b)
    if (b > a)
      b = b - a;
    else
      a = a - b;
  return a;
}
```

	TO	TO
1	2	
2	3	6
3	4	5
4	2	
5	2	1
6	EXIT	

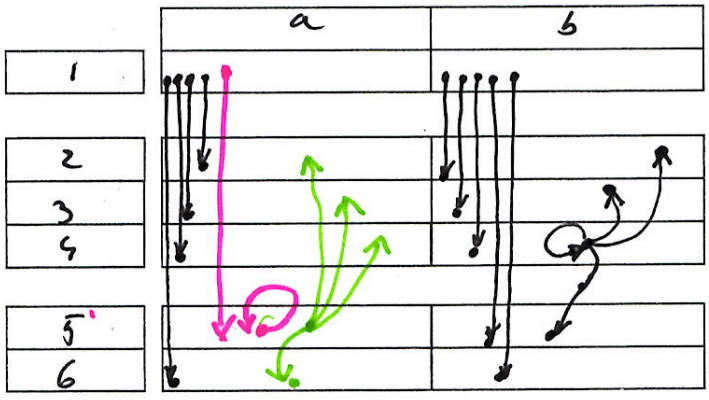
Data R/W

```
unsigned gcd(unsigned a,
             unsigned b){
  while (a != b)
    if (b > a)
      b = b - a;
    else
      a = a - b;
  return a;
}
```

	a	b
1	W	W
2	R	R
3	R	R
4	R	W, R
5	W R	R
6	R	

Data Edges

```
unsigned gcd(unsigned a,
             unsigned b){
  while (a != b)
    if (b > a)
      b = b - a;
    else
      a = a - b;
  return a;
}
```



Control Edges/ Data R/W

```
unsigned modulo(unsigned x) {
  unsigned r, q, z;
  r = x & 255;
  q = x >> 8;
  do {
    do {
      r = r + (q * 77) & 255;
      q = (q * 77) >> 8;
    } while (q != 0);
    q = r >> 8;
    r = r & 255;
  } while (q != 0);
  z = (r >= m) ? r - m : r;
  return z;
}
```

FROM	TO
1	2
2	3
3	4
4	5
5	6
6	4,7
7	8
8	9
9	4,10
10	11
11	EXIT

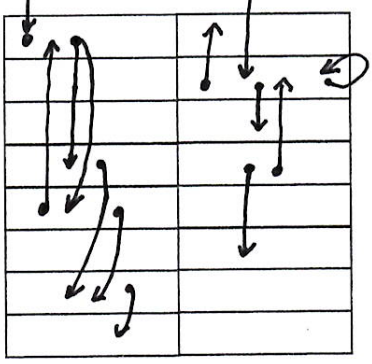
x	R	Q	Z
W			
R	W		
R		W	
	W,R	R	
		W,R	
		R	
	R	W	
	W,R		
		R	
	R		W
			R

Data Edges

```
unsigned modulo(unsigned x) {
  unsigned r, q, z;
  r = x & 255;
  q = x >> 8;
  do {
    do {
      r = r + (q * 77) & 255;
      q = (q * 77) >> 8;
    } while (q != 0);
    q = r >> 8;
    r = r & 255;
  } while (q != 0);
  z = (r >= m) ? r - m : r;
  return z;
}
```

1	2
2	3
3	4
4	5
5	6
6	4,7
7	8
8	9
9	4,10
10	11
11	EXIT

R	Q



Hardware Implementation

```
unsigned modulo(unsigned x) {  //  1
  unsigned r, q, z;

  r = x & 255;                 //  2
  q = x >> 8;                  //  3

  ------------------------------------
  do {
    do {
      r = r + (q * 77) & 255;  //  4
                               //
      q = (q * 77) >> 8;       //  5
                               //

  ------------------------------------
    } while (q != 0);          //  6


  ------------------------------------
    q = r >> 8;                //  7
    r = r & 255;               //  8

  ------------------------------------
  } while (q != 0);            //  9


  ------------------------------------

  z = (r >= m) ? r - m : r;    // 10
                               //

  return z;                    // 11
}
```

FSM DP

LOAD

COMPUTE1

INNER

COMPUTE2

OUTER

ADJ