*Figure 1*

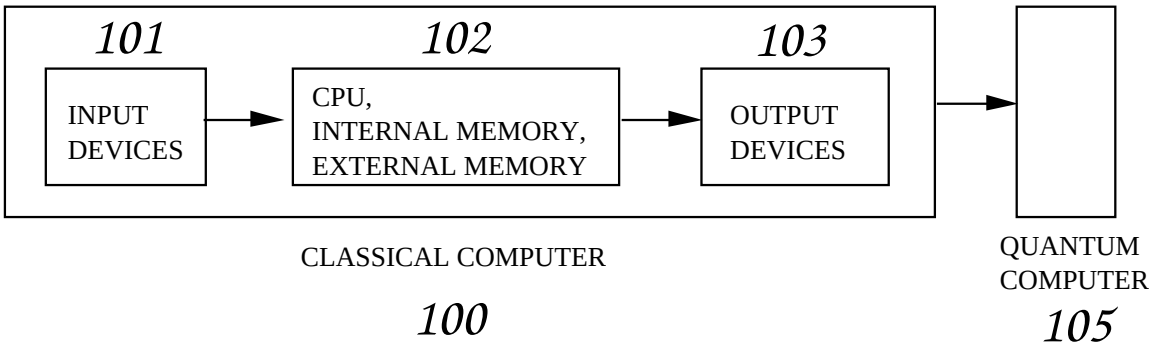
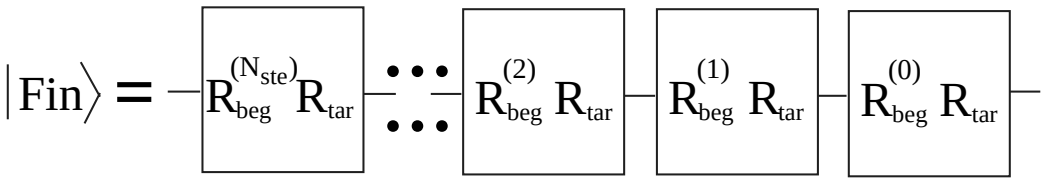

101
102
103
INPUT DEVICES
CPU, INTERNAL MEMORY, EXTERNAL MEMORY
OUTPUT DEVICES
CLASSICAL COMPUTER
100
QUANTUM COMPUTER
105

*Figure 2*




*201*

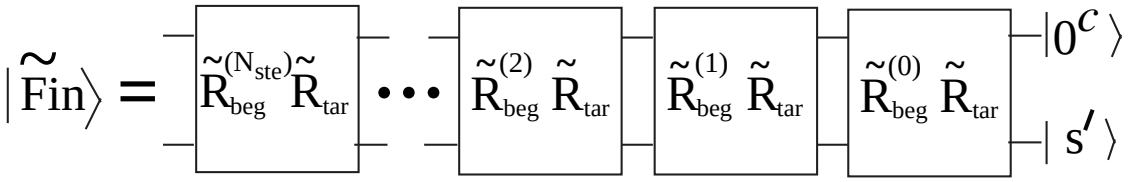
$$R_{beg}^{(j)} = \exp(i\alpha_j |s'\rangle\langle s'|) \qquad 202$$

$$R_{tar} = \exp(i\Delta\lambda |t\rangle\langle t|) \qquad 203$$

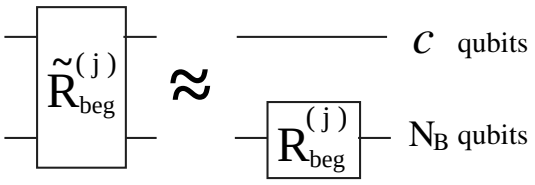
$$\gamma = 2\arccos(|\langle s'|t\rangle|) \qquad 204$$

$$e^{i\alpha}|Fin\rangle \approx |t\rangle \qquad 205$$

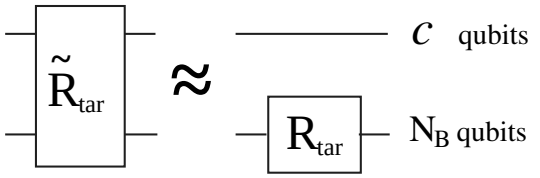
*Figure 5*




*501*




*502*




*503*

$$e^{i\alpha}|\tilde{\mathrm{Fin}}\rangle \approx \begin{matrix} -|0^c\rangle \\ -|\ t\ \rangle \end{matrix}$$

*504*

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% afga.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

g0_degs = 27 %g0 stands for gamma
del_lam_degs = 90
num_steps = 21

fi = fopen ("afga.txt", "w", "native");
fprintf(fi, "gamma(degs) = %10.4e\n", g0_degs);
fprintf(fi, "del_lam(degs) = %10.4e\n", del_lam_degs);
fprintf(fi, "num_steps = %i\n", num_steps);

g0 = (pi/180)*g0_degs;
del_lam = (pi/180)*del_lam_degs;
sg0 = sin(g0);
cg0 = cos(g0);
slam = sin(del_lam);
clam = cos(del_lam);
vz = [0;0;1];
vs0 = [sg0;0;cg0];

vs = vs0;
vr = afga_rot(vz, -del_lam, vs);
g_in = g0;
[g_out, alpha] = afga_step(g_in, g0, del_lam);

fprintf(fi,"j\tgam_j(degs)\talp_j(degs)\tvr_x\tvr_y\tvr_z\tvs_x\tvs_y\tvs_z\n");

for j=0:num_steps
    fprintf(fi,"%i\t", j);
    fprintf(fi,"%10.4e\t", g_in*180/pi);
    fprintf(fi,"%10.4e\t", alpha*180/pi);
    fprintf(fi,"%10.4e\t%10.4e\t%10.4e\t", vr(1), vr(2), vr(3));
    fprintf(fi,"%10.4e\t%10.4e\t%10.4e\n", vs(1), vs(2), vs(3));

    vs = afga_rot(vs0, -alpha, vr);
    vr = afga_rot(vz, -del_lam, vs);
    g_in = g_out;
    [g_out, alpha] = afga_step(g_in, g0, del_lam);
endfor
fclose(fi);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% afga_rot.m
%This function is called by afga.m
%It rotates a vector r
%by an angle theta about the axis a.
%r and a must be unit R^3 vectors.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function r_new = afga_rot(a, theta, r)
r_new = a*(a'*r)+ cross(a, r)*sin(theta) +(r - (a'*r)*a)*cos(theta);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% afga_step.m
%This function is called by afga.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [g_out, alpha] = afga_step(g_in, g0, del_lam)

cgin = cos(g_in);
sgin = sin(g_in);
cg0 = cos(g0);
sg0 = sin(g0);
cg0minus = cos(g0 - g_in);
sg0minus = sin(g0 - g_in);
clam = cos(del_lam);
slam = sin(del_lam);
rjs = cg0*cgin + sg0*sgin*clam;

angle_rjs = atan2(sqrt(1-rjs^2), rjs);
dg = -g0 + g_in + angle_rjs;
g_out = g_in - dg;

%alpha = asin(slam)*sgin)/sin(angle_rjs))
%this expression for alpha only works sometimes, because
%x and pi - x both have same sine

x = sin(dg)*(cgin*sgin*(1-clam) + sg0minus*rjs) +
    cos(dg)*(cgin^2 +sgin^2*clam - cg0minus*rjs);
y = sin(g0 -g_in + dg)*sgin*slam;
alpha = atan2(y,x);
```