Corresponding Manuscript: Conditional Operation of Hole Spin Qubits above 1 K — [Notebook: S3_Two_Tone_vs_Jeff_v160925_v1.nb]
Uploaded/curated in current form by (blame): M.J. Carballido | ORCID: https://orcid.org/0000-0001-7385-8284
Date: Oct. 4th, 2025

Project mirror: https://github.com/carmig00/Publications-OpenAccess-Code-Conditional2Q

How to run: Evaluate top-to-bottom (Evaluation → Evaluate Notebook).
Outputs: HDF5 (.h5) written to ./exports.

Figure mapping: This notebook reproduces the simulation of how different values for J_eff (ranging from 0-80 MHz) manifest in the two tone spectroscopy experiment. We conclude that J_eff is best represented by the difference in minima of the two-tone traces.

Comment/Note:
-The fast qubit (high f_Rabi), at lower Larmor frequency, is the one located physically on the right and is represented by pink/purple color tones, here qubit 1.
- The slow qubit (low f_Rabi), at higher Larmor frequency, is the one located physically on the left and is represented by orange/yellow color tones, here qubit 2.

```
(* Parameters *)

(* f_Rabi, MHz *)

(* We assume this to be the bare Rabi frequency of the two ideally isolated
 spin qubits. Sometimes also referred to as bare Rabi frequency,for J=0 *)
fR1t = 24;
fR2t = 13;

(* NOT USED HERE (because burst time is swept in Chevron):
 Experimentally chosen pulse durations to induce a Pi/2 flip for qubit 1 and qubit 2,
see supplementary material for details on the calibration *)
(* units in \mus *)
Tp1 = 20 × 10^-3; (* corresponding to 20 ns *)
Tp2 = 30 × 10^-3; (* corresponding to 30 ns *)

(* Larmor frequencies, 2*Pi*MHz *)
(*NOTE: these values for Larmor frequencies were chosen as a
  coarse guess by orienting ourselves on the center position of the EDSR
  peaks which are not the most accurate representation of the precise
  Larmor frequency (due to convolution of two Chevrons). In addition,
the values were rounded down to the next 10 MHz. Only after,
in the simulation of the 2-tone experiment we could more precisely narrow-
```

```
 down and adjust the larmor frequencies as fit parameters. For seeing the effect of
  J_eff on an ideal 2-tone experiment, this choice was sufficiently accurate.*)
ωz1 = 2 Pi 2090;
ωz2 = 2 Pi 2280;

(*To simulate noise, we averaged over different Larmor
 frequencies sampled from a random, normal distributed set.*)
(*sigma of normal distributuion 2*Pi*MHz *)
σωz1 = 25 × 2 Pi;
σωz2 = 25 × 2 Pi;

(* Resolution of the array of random Larmor frequencies MHz*)
resωz1 = 150;
resωz2 = 150;

(* Arrays of random Larmor freqs. MHz*)
ωz1arr = Abs[RandomVariate[NormalDistribution[ωz1, σωz1], {resωz1}]];
ωz2arr = Abs[RandomVariate[NormalDistribution[ωz2, σωz2], {resωz2}]];

(* Values of effective exchange interaction J_eff *)
(* largest value of J (we start from J_eff = 0) *)
Jmax = 80;
(* resolution of J *)
resJ = 40;
(*array of J_eff to be used in simulation*)
Jarr = Table[jval, {jval, 0, Jmax, Jmax / (resJ - 1)}];

(*Resolution of frequency scanned (x-axis) MHz*)
resω = 160;
(*initial freq MHz*)
ωin = 2 π 1800;
(*final freq MHz*)
ωfin = 2 π 2600;
(*array of freqs. scanned over, MHz*)
ωarr = Table[t, {t, ωin, ωfin, (ωfin - ωin) / (resω - 1)}];


(* Main functions *)



θso = 0;
φ1 = 0;
φ2 = 0;

ψ0a = {1, 0, 0, 0};
```

```
ψ0b = {0, 0, 0, 1};
resTg = 100;
(*initial time*)
Tg0 = 0;
(*final time, 66.6ns*)
Tgf = 3 * 1 / 30;
Tgarr = Table[t, {t, Tg0, Tgf, (Tgf - Tg0)/(resTg - 1)}];


{nx, ny, nz} = {1, 0, 0};
Jpar[J_] = 2 Pi J (nz^2 + (1 - nz^2) Cos[θso]);
(*λx00=2 Pi 30; (*Russ/Burkard resonance condition for driving*)*)
HQ[ωz1in_, ωz2in_] = KroneckerProduct[PauliMatrix[3] ωz1in/2, PauliMatrix[0]] +
   KroneckerProduct[PauliMatrix[0], PauliMatrix[3] ωz2in/2];
Rso = {
   {nx^2 + (1 - nx^2) Cos[θso],
    nx ny - nx ny Cos[θso] - nz Sin[θso], nx nz - nx nz Cos[θso] + ny Sin[θso]},
   {nx ny - nx ny Cos[θso] + nz Sin[θso], ny^2 + (1 - ny^2) Cos[θso],
    ny nz - ny nz Cos[θso] - nx Sin[θso]}, {nx nz - nx nz Cos[θso] - ny Sin[θso],
    2 Sin[θso/2] (nx Cos[θso/2] + ny nz Sin[θso/2]), nz^2 + (1 - nz^2) Cos[θso]}
  };
HJ[J_] =
  J/4 Sum[KroneckerProduct[PauliMatrix[i], (Rso.Array[PauliMatrix, 3])〚i〛], {i, 1, 3}];




(*turn on time of pulses*)
dt0 = 0.001;
(*wait time between pulses*)
Tw = 0.001;

(*driving amplitudes,here normalised to 2 Pi and equal to each other.One could choose to
 e.g.drive one qubit harder than the other one.(this could lead to over-fitting)*)
λx10 = 2 Pi; (*Drive amplitude Q1*)
λx20 = 2 Pi; (*Drive amplitude Q2*)

(*first pulse for a time Tflip1*)
```

```
λx1[t_] = λx10 (1/2 + 1/2 Erf[(Tflip1 - t) / dt0]);
(*second pulse after a time Tflip1+Tw*)
λx2[t_] = λx20 (1/2 + 1/2 Erf[(t - Tflip1 - Tw) / dt0]);

(* Drive Hamiltonian *)
Hλ[t_] = λx1[t] Cos[ωx1 t + φ1]
    (fR1 KroneckerProduct[{nx, ny, nz}.Array[PauliMatrix, 3], PauliMatrix[0]] + fR2
       KroneckerProduct[PauliMatrix[0], {nx, ny, nz}.Array[PauliMatrix, 3]]) + λx2[t] Cos[
     ωx2 t + φ2] (fR1 KroneckerProduct[{nx, ny, nz}.Array[PauliMatrix, 3], PauliMatrix[0]] +
      fR2 KroneckerProduct[PauliMatrix[0], {nx, ny, nz}.Array[PauliMatrix, 3]]);

(*Total Hamiltonian*)
Htot[t_, J_, ωz1in_, ωz2in_] = HQ[ωz1in, ωz2in] + HJ[J] + Hλ[t];
ρSf =
  Function[{ψ0, ψro, Tflip1in, Tflip2in, ωz1in, ωz2in, ωx1in, ωx2in, fR1in, fR2in, Jin}, sub =
    {Tflip1 → Tflip1in, Tflip2 → Tflip2in, ωx1 → ωx1in, ωx2 → ωx2in, fR1 → fR1in, fR2 → fR2in};
   Tfin = Tflip1 + Tw + Tflip2 /. sub;
   ψfin[t_] = {a[t], b[t], c[t], d[t]} /. NDSolve[
       {Htot[t, Jin, ωz1in, ωz2in].{a[t], b[t], c[t], d[t]} ==
          ⅈ D[{a[t], b[t], c[t], d[t]}, t] /. sub,
        a[0] == ψ0⟦1⟧, b[0] == ψ0⟦2⟧, c[0] == ψ0⟦3⟧, d[0] == ψ0⟦4⟧},
       {a[t], b[t], c[t], d[t]}, {t, 0, Tfin},
       Method → "PDEDiscretization" → {"MethodOfLines",
          "SpatialDiscretization" → "FiniteElement", MaxCellMeasure → 0.0005 / fR1}]⟦1⟧;
   (*readout signal*)
   Sum[Abs[(ψro⟦i⟧.ψfin[Tfin])]^2, {i, 1, Dimensions[ψro]⟦1⟧}]];
```

```
(*Two-tone spectroscopy vs J, averaged over different Larmor ferquencies*)
(* The four frequencies f1 to f4 are chosen to represent
 the experimentally "ideal" case, where: ωz1(2)± Jpar[Jval]/2 *)
(* We read out in the 0,1,0,0 and *)
(* 0,0,1,0 states (anti-parallel) which are not blocked in PSB *)
(*NOTE: Below, we apply the Pi pulse Tp1 for f1 and f2 and Tp2 for f3 and f4. *)

resF1vsJ =
  ParallelTable[(ρSf[ψ0a, {{0, 1, 0, 0}, {0, 0, 1, 0}}, Tp1, Tp1, ωz1arr⟦j⟧, ωz2arr⟦j⟧,
       ωz1 - Jpar[Jarr⟦k⟧] / 2, ωarr⟦i⟧, fR1t, fR2t, Jpar[Jarr⟦k⟧]] +
      ρSf[ψ0b, {{0, 1, 0, 0}, {0, 0, 1, 0}}, Tp1, Tp1, ωz1arr⟦j⟧, ωz2arr⟦j⟧,
       ωz1 - Jpar[Jarr⟦k⟧] / 2, ωarr⟦i⟧, fR1t, fR2t, Jpar[Jarr⟦k⟧]]) / 2,
   {j, 1, resωz1}, {i, 1, resω}, {k, 1, resJ}]; // AbsoluteTiming
resF2vsJ =
  ParallelTable[(ρSf[ψ0a, {{0, 1, 0, 0}, {0, 0, 1, 0}}, Tp1, Tp1, ωz1arr⟦j⟧, ωz2arr⟦j⟧,
       ωz1 + Jpar[Jarr⟦k⟧] / 2, ωarr⟦i⟧, fR1t, fR2t, Jpar[Jarr⟦k⟧]] +
      ρSf[ψ0b, {{0, 1, 0, 0}, {0, 0, 1, 0}}, Tp1, Tp1, ωz1arr⟦j⟧, ωz2arr⟦j⟧,
       ωz1 + Jpar[Jarr⟦k⟧] / 2, ωarr⟦i⟧, fR1t, fR2t, Jpar[Jarr⟦k⟧]]) / 2,
   {j, 1, resωz1}, {i, 1, resω}, {k, 1, resJ}]; // AbsoluteTiming
resF3vsJ =
  ParallelTable[(ρSf[ψ0a, {{0, 1, 0, 0}, {0, 0, 1, 0}}, Tp2, Tp2, ωz1arr⟦j⟧, ωz2arr⟦j⟧,
       ωz2 - Jpar[Jarr⟦k⟧] / 2, ωarr⟦i⟧, fR1t, fR2t, Jpar[Jarr⟦k⟧]] +
      ρSf[ψ0b, {{0, 1, 0, 0}, {0, 0, 1, 0}}, Tp2, Tp2, ωz1arr⟦j⟧, ωz2arr⟦j⟧,
       ωz2 - Jpar[Jarr⟦k⟧] / 2, ωarr⟦i⟧, fR1t, fR2t, Jpar[Jarr⟦k⟧]]) / 2,
   {j, 1, resωz1}, {i, 1, resω}, {k, 1, resJ}]; // AbsoluteTiming
resF4vsJ =
  ParallelTable[(ρSf[ψ0a, {{0, 1, 0, 0}, {0, 0, 1, 0}}, Tp2, Tp2, ωz1arr⟦j⟧, ωz2arr⟦j⟧,
       ωz2 + Jpar[Jarr⟦k⟧] / 2, ωarr⟦i⟧, fR1t, fR2t, Jpar[Jarr⟦k⟧]] +
      ρSf[ψ0b, {{0, 1, 0, 0}, {0, 0, 1, 0}}, Tp2, Tp2, ωz1arr⟦j⟧, ωz2arr⟦j⟧,
       ωz2 + Jpar[Jarr⟦k⟧] / 2, ωarr⟦i⟧, fR1t, fR2t, Jpar[Jarr⟦k⟧]]) / 2,
   {j, 1, resωz1}, {i, 1, resω}, {k, 1, resJ}]; // AbsoluteTiming

Export["C:\\Users\\export\\two_tone_vs_J_average_Larmor_sigma" <> TextString[σωz1] <>
  "MHz_J" <> TextString[Jval] <> "MHz_vDDMMYY_vX.h5", {resF1vsJ, resF2vsJ, resF3vsJ,
  resF4vsJ, ωz1arr / (2 Pi), ωz2arr / (2 Pi), ωarr / (2 Pi), Jarr}, {"Datasets",
  {"resF1vsJ", "resF2vsJ", "resF3vsJ", "resF4vsJ", "fR1arr", "fR2arr", "farr", "Jarr"}}];
Out[ ]=
{5491.59, Null}
Out[ ]=
{5476.5, Null}
Out[ ]=
{8083.16, Null}
Out[ ]=
{7948.77, Null}
```